

Niko Rudnäs

## **Interaktiivisen 3D real-time web-sovelluksen vienti Oculus Rift -virtuaalilaseille.**

CASE: MyHome

Opinnäytetyö

Kevät 2015

SeAMK Tekniikka

Tietoverkkotekniikka/Ohjelmistotekniikka Tutkinto-ohjelma



SEINÄJOEN AMMATTIKORKEAKOULU  
SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

SEINÄJOEN AMMATTIKORKEAKOULU

## Opinnäytetyön tiivistelmä

Koulutusyksikkö: Tekniikan yksikkö

Tutkinto-ohjelma: Tietotekniikka

Suuntautumisvaihtoehto: Tietoverkkotekniikka

Tekijä: Niko Rudnäs

Työn nimi: Interaktiivisen 3D real-time web-sovelluksen vienti Oculus Rift -virtuaalilaseille.

Ohjaaja: Hilkka Niemelä

Vuosi: 2015

Sivumäärä: 54

Liitteiden lukumäärä: 0

---

Opinnäytetyö toteutettiin 3D Avenue International Oy:n MyHome-web-sovellukseen liittyen. 3D Avenue on teknologia-alan start-up-yritys, jonka päätoimisiin toimialoihin kuuluvat ohjelmistojen suunnittelu ja toteutus sekä graafinen suunnittelu ja toteutus.

Tämän opinnäytetyön tarkoituksena oli suunnitella ja toteuttaa Oculus Rift -virtuaalilasien integrointi osaksi MyHome-sovellusta, sekä luoda toimiva tapa liikua sovelluksessa.

Työssä teoriaosuudessa esiteltiin virtuaalitodellisuuden tavoitteita ja mahdollisuuksia sekä interaktiivisten web-sovellusten luontiin käytettävien tekniikoiden periaatteita. Käytännön osuudessa esitettiin kuinka on mahdollista suunnitella ja toteuttaa web-sovelluksen vienti selaimesta Oculus Rift -virtuaalilaseille, muotoilla se lasissa näkyväksi 3D-ympäristöksi sekä luoda sopivat kontrollit tähän asetukseen. Tarkoituksena oli myös löytää mahdollisimman käyttäjäystävällinen tapa, joka ei vaatisi ylimääräisten lisäosien asennusta käyttäjän laitteelle.

Työn lopputuloksena syntyi toimiva tapa muotoilla ja siirtää HTML5-canvas-elementti korrektisti näkyväksi 3D-ympäristöksi Oculus Rift Development Kit 2 -virtuaalilaseille. Lisäksi luotiin toimiva kontrollointitapa sovelluksessa liikkumiseen.

Avainsanat: Verkko-ohjelmointi, virtuaalitodellisuus, oculus rift, responsive web design, 3D-mallinnus

SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

## **Thesis abstract**

Faculty: School of Technology

Degree programme: Information Technology

Specialisation: Information Network Technology

Author: Niko Rudnäs

Title of thesis: Exporting the interactive 3D real-time web application to Oculus Rift -virtual reality headset.

Supervisor: Hilkka Niemelä

Year: 2015                      Number of pages: 54      Number of appendices: 0

---

This thesis was commissioned by 3D Avenue Oy. 3D Avenue Oy is a start-up company which develops software design and graphical implementations.

The aim of this thesis was to plan and implement the integration of Oculus Rift -virtual reality headset with the 3D Avenue Oy's MyHome -web application. Another aim was to create a movement control system to be used with the virtual reality headset.

In the theory part of this thesis are presented the aims and possibilities of virtual reality, and the principles of the technologies used to create an interactive web application. In the practical part is presented a method used for exporting a web application that is running on HTML5 canvas element to the Oculus Rift -virtual reality headset. The aim was also to create as user-friendly a way as possible.

As the result of this thesis a practical method for exporting the HTML5 canvas element to the Oculus Rift -virtual reality headset using Javascript programming language was created. Also a way to smoothly control the camera and viewport rotation was implemented into the application used in this thesis.

Keywords: virtual reality, oculus rift, responsive web design, web application

## SISÄLTÖ

Opinnäytetyön tiivistelmä.....	1
Thesis abstract.....	2
SISÄLTÖ.....	3
Kuva-, kuvio- ja taulukkoluettelo.....	5
Käytetyt termit ja lyhenteet .....	6
1 JOHDANTO .....	9
1.1 Työn tausta .....	9
1.2 Työn tavoite .....	9
1.3 Työn rakenne .....	10
1.4 Toimeksiantajan esittely.....	10
2 VIRTUAALITODELLISUUS JA KOLMIULOTTEINEN GRAFIIKKA	11
2.1 Virtuaalitodellisuus kokemuksena .....	11
2.2 Virtuaaliympäristön rakentaminen.....	13
2.3 Virtuaalitodellisuuden tulevaisuus .....	14
3 3D-MALLINNUS.....	16
3.1 Kolmiulotteinen tietokonegrafiikka.....	16
3.2 3D-mallien luonti .....	16
3.3 3D-grafiikka internetselaimessa .....	19
4 VIRTUAALIYMPÄRISTÖJEN VÄLINEITÄ.....	21
4.1 Virtuaalitodellisuuslasit.....	21
4.2 3D-kamera .....	23
4.3 Peliohjain .....	24
5 KÄYTETYT OHJELMAT JA TEKNOLOGIAT .....	25
5.1 Web-ohjelmointi .....	25
5.1.1 HTML5.....	25
5.1.2 CSS 3 .....	26
5.1.3 Javascript.....	27
5.1.4 PHP.....	28
5.2 Teknologiat .....	29
5.2.1 OpenGL .....	29

5.2.2	Canvas-elementti.....	30
5.2.3	AJAX.....	31
5.2.4	Websocket.....	31
5.2.5	WebGL.....	32
5.3	Apuvälineet .....	34
5.3.1	Three.js.....	35
5.3.2	VRrender.js.....	36
5.3.3	VR Enabled Chrome .....	36
5.3.4	XAMPP .....	37
6	MYHOME.....	38
6.1	Näkymät.....	38
6.2	Tekniikoiden valinta .....	41
6.3	Toteutus.....	42
6.4	Ongelmat .....	45
7	YHTEENVETO JA POHDINTA .....	48
	LÄHTEET .....	50
	LIITTEET .....	54

## Kuva-, kuvio- ja taulukkoluettelo

Kuvio 1. Virtuaaliympäristö virtuaalikuutiassa (Telepresenceoptions 2012.).....	12
Kuvio 2. Sotilaskoulutus virtuaalisesti. (Wired 2012.).....	14
Kuvio 3. Grafiikkaputken vaiheet (The Pipeline Behind Modeling and Animating a Game Character in Zelgor 2013.) .....	18
Kuvio 4. Vanha OpenGL-tekniikalla toteutettu peli (Youtube 2015.).....	19
Kuvio 5. Oculus Rift avattuna (iFixit 2014.) .....	22
Kuvio 6. Nimble VR kiinnitettynä Oculus Riftissä (Gilbert 2014.) .....	23
Kuvio 7. Virtuix Omni -laite (Digitalized technologies [Viitattu 5.3.2015].) .....	24
Kuvio 8. WebGL:llä toteutettua selaingrafiikkaa (Tarun 2013.) .....	34
Kuvio 9. Laseissa näkyvä kuva imitoidaan tietokoneen näytölle.....	36
Kuvio 10. MyHome 3D-näkymä. ....	38
Kuvio 11. Ylänäkymä. ....	39
Kuvio 12. Puolitettu näkymä. ....	40
Kuvio 13. Muotoiltu näkymä. ....	41
Kuvio 14. Kirjaston lisäys projektiin.....	42
Kuvio 15. Virtuaalilaitteen haku ja tunnistus.....	43
Kuvio 16. Rendereiden luonti.....	44
Kuvio 17. Renderöinti.....	44
Kuvio 18. Kameran kontrollointi .....	45
Kuvio 19. Liikkuminen .....	46

## Käytetyt termit ja lyhenteet

**2D** Two-dimensional eli kaksiulotteinen. Kappaleella voi olla kaksi ulottuvuutta.

**3D** Three-dimensional eli kolmiulotteinen. Kappaleella voi olla kolme ulottuvuutta.

### Alustariippumattomuus

Tarkoittaa ohjelmointikieltä tai sovellusta, joka ei ole sidoksissa tiettyyn laitteistoalustaan tai käyttöjärjestelmään.

**Bloatware** Turhaan esiasennettu tai ohjelman mukanaan asentama sovellus.

**Chroma** Värin värisävyn tarkkuus, joka määräytyy siitä, paljonko siinä on mustaa tai valkoista

**CPU** Central Processing Unit eli suoritin on tietokoneen osa, joka suorittaa konekielisiä käskyjä.

**DOM** Document Object Model. Ohjelmointirajapinta, joka mahdollistaa merkkauškielen dokumenttien sisällön muokkauksen.

**Elementti** Tässä opinnäytetyössä elementillä tarkoitetaan verkkosivun osaa, joka on jäsennetty osaksi Document Object Modelia (DOM).

**Eväste** Evästä eli cookie on dataa, jonka web-palvelin tallentaa käyttäjän tietokoneelle. Sisältää yleensä tietoa vierailluista internet-sivustoista.

<b>Formaatti</b>	Ilmaisee tallennetun tiedoston tyyppin eli rakenteen. Monissa tiedostomuodoissa on olemassa julkinen määritelmä, jossa kuvataan miten tieto tulee koodata.
<b>Funktio</b>	Funktio eli aliohjelma on ohjelmoinnissa itsenäinen ohjelman osa, joka suorittaa tietyn toiminnon ja jota voidaan kutsua eri puolilta ohjelmakoodia.
<b>GPU</b>	Graphic Processing Unit eli grafiikkaprosessori on tietokoneen osa, joka suorittaa 2D- tai 3D-grafiikan renderöintiä, joka muuten jäisi suorittimen laskettavaksi.
<b>Immersiivinen</b>	Virtuaalitodellisuuteen upottava. Syventää kokemuksen niin, että keskittyminen on kokonaan pelissä.
<b>Integrointi</b>	Jonkin asian tuomista osaksi jotakin toista asiaa.
<b>Interaktiivinen</b>	Vuorovaikutteinen tietoteknisen laitteen käyttö, jossa käyttäjä ”keskustelee” laitteen kanssa antaen käskyjä ja saa palautteen heti.
<b>Kickstarter</b>	Joukkorahoituspalvelu, joka keskittyy erityisesti luovien projektien joukkorahoitukseen.
<b>Merkkauskieli</b>	Merkkauskieli on formaali kieli, jolla kuvataan tekstin rakennetta tai esitystapaa liitännäistietona.
<b>Plugin</b>	Plugin eli liitännäinen on tietokoneohjelma, joka toimii vuorovaikutuksessa isäntäsovelluksen kanssa tarjotakseen tietyn toiminnon tarvittaessa.
<b>Rajapinta</b>	Rajapinta eli API (Application Programming Interface) mahdollistaa eri ohjelmien vuorovaikutuksen keskenään.
<b>Real-time</b>	Reaaliaikainen eli ajantasainen tapahtuma.



<b>Renderöinti</b>	Kuvan luominen mallista tietokoneohjelman avulla. Malli on yleensä datatiedosto, joka sisältää tiedot mm. geometriasta, katselukulmasta, tekstuurista sekä valaistuksesta.
<b>Skriptikieli</b>	Skriptikieli eli komentosarjakieli on tapa, jolla kirjoitetaan skriptejä, jotka automatisoivat tehtäviä ilman että tarvitaan varsinaisia ohjelmointikieliä.
<b>Stereoskooppinen</b>	3D-kuva, joka renderöidään kahteen kertaan, vasemmalle sekä oikealle silmälle ja synkronoidaan yhtenäiseksi.
<b>Syntaksi</b>	Varattujen sanojen ja lauseiden tunnistus ohjelmakoodissa.
<b>Tekstuuri</b>	Kaksiulotteinen kuva, jota käytetään kolmiulotteisen pinnan kuvioimiseen.
<b>Vektorigrafiikka</b>	Tietokonegrafiikka, jossa objektien muodot ja ominaisuudet esitetään koordinaatein sekä matemaattisin funktioin.
<b>Vr</b>	Virtual reality eli virtuaalitodellisuus.
<b>Web</b>	Lyhennys sanasta "World Wide Web" eli internet.

# 1 JOHDANTO

## 1.1 Työn tausta

Vielä muutama vuosi sitten virtuaalitodellisuus oli outo ja kaukainen käsite monelle. Kuitenkin uusien tekniikoiden avulla on melkein kaikilla mahdollisuus hankkia kohtuullisen hintaisia laitteita, jotka tuovat virtuaalitodellisuuden omaan kotiin. Kyse ei ole hologrammiteknologiasta, vaan näyttöpaneeleista ja ohjelmistoista, jotka tukevat kuvien muuntamisen ja näyttämisen molemmille silmille omina kuvina, soveltaen 3D-elokuvissa käytettävää tekniikkaa. Uusimmat matkapuhelimet soveltuvat myös hyvin virtuaalitodellisuuskäyttöön.

Virtuaalitodellisuudella tarkoitetaan tietokonesimulaation avulla luotua keinotekoisia ympäristöä. Ympäristö voidaan luoda jostakin todellisesta tai täysin kuvitteellisesta ympäristöstä. Tämä mahdollistaa esimerkiksi sen, että rakennusten ja asioiden mahdollista ulkonäköä ja toimintaa voi tutkia etukäteen mallintamalla haluttu asia tai esine.

Kun toiminta ja ulkonäkö yhdistetään, syntyy tekniikka nimeltään kolmiulotteinen virtuaaliympäristö. Tähän voi vielä yhdistää lisätoimintoja kuten interaktiivisuus, sekä reaaliaikainen tapahtumien käsittely. Lopputuloksena voi olla vaikka henkilökohtainen kerrostalo halutussa ympäristössä, jossa on mahdollista liikkua hissillä kerrosten välillä ja istua alas halutussa huoneessa katsomaan vaikka televisiota.

## 1.2 Työn tavoite

Tässä insinöörityössä käytiin läpi tekniikoita, joita tarvitaan 3D-mallien toteuttamiseen sekä näyttämiseen web-ympäristössä ja virtuaalitodellisuudessa. Lisäksi tutkittiin, onko mahdollista luoda MyHome-sovelluksesta kolmiulotteinen virtuaalitodellisuus Oculus Rift -laseihin yhteensopivaksi, ja kuinka se tulisi tehdä. Laite, jota tässä opinnäytetyössä käytettiin virtuaalitodellisuuden luomiseen, oli Oculus Rift Development Kit 2 -virtuaalitodellisuuslasit.

### 1.3 Työn rakenne

Luvussa 2 kerrotaan virtuaalitodellisuudesta ja sen käyttötarkoituksista, sekä virtuaalitodellisuuden tulevaisuuden näkymiä.

Luvussa 3 perehdytään 3D-mallinnuksen periaatteisiin ja tapoihin luoda ja näyttää 3D-grafiikkaa.

Luvussa 4 tutkitaan virtuaalitodellisuuden käyttötarkoitukseen luotuja laitteita.

Viidennessä luvussa esitellään web-sovelluksien luonnissa käytettäviä teknologioita sekä työn tekijän käyttämiä välineitä työn toteutuksen luomisessa.

Luvussa 6 siirrytään työn käytännön osuuteen eli virtuaalilasien integrointiin osaksi MyHome-sovellusta. Seitsemännessä luvussa on opinnäytetyön yhteenveto, jossa pohditaan työn tuloksia ja tavoitteiden toteutumista.

### 1.4 Toimeksiantajan esittely

3D Avenue International Oy on Seinäjoen keskustassa sijaitseva kasvuyritys, joka tuottaa MyHome-virtuaalipalvelua kiinteistöjen digitaaliseen markkinointiin. MyHome on pelimäinen ja interaktiivinen virtuaaliesittelypalvelu kaikille kiinteistöille. (Mähönen 2015.)

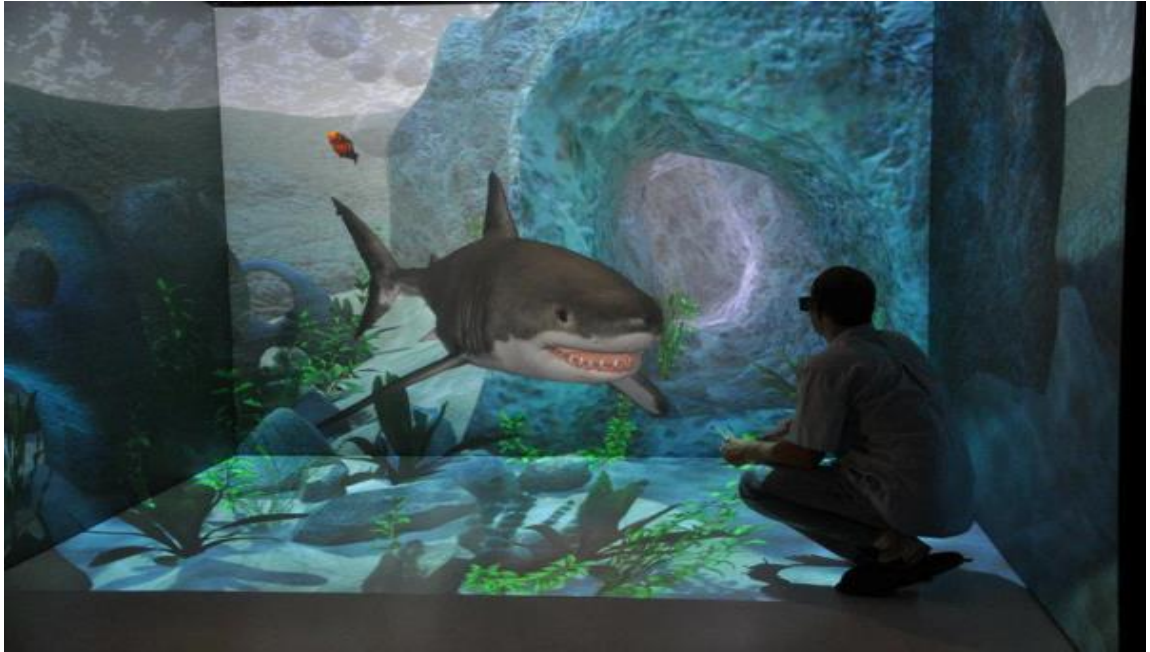
MyHome-sovelluksessa loppukäyttäjä voi liikkua vapaasti virtuaalisissa kiinteistökohteissa oman päätelaitteensa internetselaimella. Kohteista pyritään luomaan mahdollisimman todenmukaisia malleja esittelemään jo olemassa olevia tai tulevia kiinteistöjä. (Mähönen 2015.)

## 2 VIRTUAALITODELLISUUS JA KOLMIULOTTEINEN GRAFIIKKA

### 2.1 Virtuaalitodellisuus kokemuksena

Virtuaalitodellisuus (VR), jota kutsutaan joskus myös immersiiiviseksi multimedialaksi, on tietokonesimuloitu ympäristö, joka voi simuloida fyysistä läsnäoloa paikoissa reaali maailmassa tai mielikuvituksellisissa ympäristöissä. Virtuaalitodellisuus voi luoda virtuaalisia aistillisia kokemuksia, joita ovat esimerkiksi näkö, kuulo ja tunto. (Hietala 2014.)

Useimmat nykyiset virtuaalitodellisuusympäristöt ovat pääasiassa empiirisiä kokemuksia, näytettynä joko tietokoneiden näytöiltä tai käyttämällä erityisiä stereoskooppisia näyttöjä. Jotkut näistä sisältävät aistillisten tuntojen palauttamista, kuten lääketieteeseen ja sotilaalliseen käyttöön tarkoitetut virtuaaliset harjoitteluympäristöt. Lisäksi virtuaalitodellisuudessa on nykyään mahdollista eläytyä enemmän ympäristön mittasuhteisiin, sillä on kehitetty käsineitä, joita käyttämällä voi tuoda käsien liikkeet mukaan virtuaali maailmaan sekä tietynlaisia juoksumattoja, joilla voi tuoda oikean kävelemisen tai juoksemisen mukaan virtuaali maailmaan lisäten todennukaisuuden tuntua. Siltikin on vielä hankala kehittää tarkkoja, hyvin todentuntuisia virtuaali elämyksiä, sillä teknologia ei ole vielä kehittynyt tarpeeksi (kuvio 1). Kehitystä vaativat vielä prosessorin laskentateho, kuvien resoluutiot sekä tietoliikennekaistan leveys. (Hietala 2014.)



Kuvio 1. Virtuaaliympäristö virtuaalikuutiassa (Telepresenceoptions 2012.)

On mahdollista, että tulevaisuudessa pystytään katsomaan elokuvia ja televisio-ohjelmia päähän asetettavien lasien avulla kytkettynä tietokoneeseen niin että katsoja voisi tuntea olevansa kohtauksen sisällä ja toimintaa tapahtuu ympärillä. Tällaista kutsutaan virtuaaliavaruudeksi. (Hietala 2014.)

Vaikka virtuaalitodellisuus on tullut tutuksi usein elokuvaviihteen ja peliteollisuuden erikoisalana, käytetään sitä nykyisin paljon myös muissa sovelluksissa tieteenalojen saralla. Virtuaalitodellisuutta käytetään teollisuudessa usein tarkastelemaan etukäteen, miten jokin asia toimii tai miltä se näyttää, tällöin säästetään valtavasti aikaa ja vaivaa. Suunnittelijat voivat myös käyttää virtuaalitodellisuutta mallintamaan kaupunkimiljöön näkymää ja kokeilla, miten suunnitelmat toimivat käytännön tasolla. Tämän avulla voidaan tutkia erilaisia tilavaikutelmia ja testata etukäteen, miltä arkkitehtien uudisrakennushankkeet näyttäisivät. (Hakala 2014.)

## 2.2 Virtuaaliympäristön rakentaminen

Sanalla virtuaaliympäristö voi olla kaksi eri merkitystä. Sillä tarkoitetaan tietoteknillisten laitteiden muodostamaa verkostoa, jossa ne voivat kommunikoida keskenään ilman ulkopuolisia yhteyksiä. Virtuaaliympäristöllä voidaan myös tarkoittaa virtuaalimallinnettua tilaa, jonka tarkoituksena on luoda käyttäjälle immersion tunteen. Tässä työssä termillä tarkoitetaan jälkimmäistä asiaa.

Tässä opinnäytetyössä käytännön osuudessa tutkittava MyHome-sovellus on virtuaaliympäristö. Sovelluksen sisältö on luotu pääosin Blender-nimisellä 3D-mallinnusohjelmalla, johon on selaimeen tuomisen jälkeen lisätty toimintoja Javascript-koodikielellä.

Yleensä virtuaaliympäristön mallintaminen aloitetaan valitsemalla ohjelma, jolla mallinnus suoritetaan. Kuitenkin monesti ammattimaisien ympäristöjen luomiseen käytetään useampaa eri mallinnusohjelmaan eri osioiden mallintamiseen, sillä jokaisella mallinnusohjelmalla on omat vahvuutensa ja heikkoutensa. Toiset voivat olla tehokkaita ympäristön ja maiseman luontiin kun toiset taas pienien yksityiskohtien luontiin tai animaatioiden tekemiseen. (Williamson [Viitattu 16.2.2015].)

Mallit tehdään monesti jostakin jo olemassa olevasta objektista, jolloin siitä on käsitys, kuva tai tiedot, joiden perusteella mallinnus suoritetaan. Vaikkei mallinnettavasta asiasta olisikaan tarkkoja tietoja, voi kokeilun kautta pyrkiä toteuttamaan visiota haluttavasta lopputuloksesta. Yleensä kun mallinnettava objekti on muodoltaan valmis, annetaan sille väri tai tekstuuri tuomaan realismin tunnetta. (Williamson [Viitattu 16.2.2015].)

Kun objekteja on mallinnettu tarvittava määrä, tuodaan ne samaan näkymään ja yhdistetään kokonaisuudeksi. Tässä apuna on yleensä mittaruudukko sekä kordinaattivektori, joiden avulla on helpompaa skaalata ja yhdistää objektit kokonaiseksi virtuaaliympäristöksi. Lopuksi kokonaisuus renderöidään näytettäväksi näyttöpäätteellä. (Williamson [Viitattu 16.2.2015].)

### 2.3 Virtuaalitodellisuuden tulevaisuus

Tällä hetkellä täydellistä virtuaalitodellisuuden kokemusta rajoittavat teknologian tehottomuus. Kuitenkin jo lähitulevaisuudessa tietokoneiden laskentatehot tulevat moninkertaistumaan, kumoten nämä esteet. (Lepton 2011.)

Käyttötarkoituksia tulevaisuudessa virtuaalitodellisuudelle ovat ainakin koulutus, harjoittelu, psykoterapia, lääketiede, urheilu sekä visuaalinen kommunikointi. Edellä mainituista varsinkin koulutuksen ja harjoittelun suhteen mahdollisuudet ovat loputtomat. Maantietoa olisi paljon mielenkiintoisempaa opiskella, jos olisi mahdollista käydä paikan päällä katsomassa, miltä vaikka 1600-luvun Rooma näytti. Myös sotilaalliseen käyttöön voisi soveltaa esimerkiksi taistelu- tai asekoulutusta virtuaalisesti (kuvio 2). Psykoterapian suhteen virtuaalitodellisuus toisi mahdollisuuden siirtyä hoidettavan potilaan mieleen, sekä hoitaa fobioita hitaalla ja turvallisella tavalla. (Lepton 2011.)



Kuvio 2. Sotilaskoulutus virtuaalisesti. (Wired 2012.)

Lääketieteessä virtuaalitodellisuus tulee luultavasti osaksi vammaisten potilaiden hoitoa. Sosiaaliseen kanssakäymiseen kykenemättömät voivat turvallisesti harjoitella kanssakäymistä virtuaalisesti ja raajoistaan halvaantuneet voivat harjoitella raajojen käyttöä virtuaalisesti. (Lepton 2011.)

Eri tutkimukset urheilijoista ovat osoittaneet, että pelkkä urheiluliikkeiden ajattelu mielessä on 80–85 % yhtä tehokasta kehittymistä kuin liikkeiden fyysinen harjoittelu. Tämä tarkoittaa sitä, että uneksiminen, meditointi sekä hypnoosi ovat melkein yhtä tehokkaita tapoja harjoitella, kuin tehdä fyysistä työtä. Virtuaalitodellisuus toisi varmasti suurta apua esimerkiksi vammautuneille urheilijoille, jotka silti haluavat parantaa taitojaan, vaikka olisivat liikuntakyvyttömiä. (Lepton 2011.)

Virtuaalitulevaisuudella on hyötyjä myös kommunikoinnissa. Päivittäin kommunikoidaan paljon puhelimen, Facebookin tai web-kameran avulla. Nämä eivät kuitenkaan yllä samalle tasolle kuin kasvotusten juttelu. Virtuaalitodellisuus toisi mahdollisuuden tavata yhden tai useamman henkilön samanaikaisesti valitsemassaan ympäristössä. Olisi myös mahdollista tuoda mukaan elekieli, kasvon ilmeet sekä käsien liikehdintä. (Lepton 2011.)



### 3 3D-MALLINNUS

#### 3.1 Kolmiulotteinen tietokonegrafiikka

Kolmiulotteisuus tietokonegrafiikassa tarkoittaa sitä, että esinettä tai asiaa voi tutkia kolmen akselin suunnalta. X-akseli liitetään yleensä kappaleen leveyteen, Y-akseli korkeuteen ja Z-akseli syvyyteen. Vaikka esineet mallinnetaan kolmiulotteiseksi, näytetään ne yleensä silti kaksiulotteiselta pinnalta, esim. tietokoneen näytöllä. 3D-tietokonegrafiikka ei eroa kovinkaan suuresti 2D-tietokonevektorigrafiikasta. 2D-grafiikkasovellukset voivat käyttää 3D-tekniikoita tietynlaisten efektien luontiin, kuten valot, ja taas 3D-sovellukset voivat käyttää 2D-renderöintitekniikoita. Kolmiulotteista tietokonegrafiikkaa kutsutaankin joskus 3D-malleiksi (ScienceDaily [Viitattu 27.1.2015].)

Renderöintiä lukuun ottamatta 3D-malli on tallennettuna graafisessa data-tiedostossa. Kuitenkin tässä on ongelma: 3D-malli on matemaattinen esitys josta-kin kolmiulotteisesta objektista ja se ei teknisesti ole grafiikkaa, ennen kuin se on renderöity ja visuaalisesti esitetty. (ScienceDaily [Viitattu 27.1.2015].)

#### 3.2 3D-mallien luonti

3D-mallien ja animaatioiden luonti on moniosainen prosessi, jota kutsutaan grafiikkaputkeksi. Kun ottaa vertailuun esimerkiksi animaatioelokuvien Kung Fu Pandan päähenkilön Po ja Shrek-elokuvan Aasin, voi todeta että hahmot näyttävät aivan erilaisilta, mutta kuitenkin molempien luontitapa tuotantovaiheessa on hyvin samanlainen. Asioiden ja esineiden luonti kiillotetuiksi 3D-malleiksi tai animaatioiksi käykin usein eri kohtia, jotka on esitelty seuraavassa. (kuvio 3). (Slick [Viitattu 16.2.2015].)

**Esivalmistelu:** Esivalmistelussa kuvitetaan haluttava lopputulos. Hyvä tapa on tehdä piirustuksia mielikuvasta, suunnitella väripalettia ja miettiä hieman yksityiskohtia. (Slick [Viitattu 16.2.2015].)

**3D-Mallinnus:** Kun mallinnettavasta asiasta on luotu 2D-malli vaikka paperille tai tietokoneen piirtotyökalulla, seuraavaksi siitä muunnetaan kolmiulotteinen kappale. Nykyaikaisten mallintajien grafiikkaputkiin kuuluu kaksi päätekniikkaa joilla malleja luodaan. Ensimmäinen on monikulmiomallintaminen ja toinen on digitaalinen kuvanveisto. Näillä molemmilla on omat hyvät ja huonot puolensa. Monikulmiomallintaminen sopii hyvin mekaanisiin ja arkkitehtuurisiin malleihin, kun taas digitaalinen kuvanveisto sopii paremmin orgaanisten (henkilö)mallien tekoon. (Slick [Viitattu 16.2.2015].)

**Varjostus ja kuviointi:** Seuraavana vuorossa on varjostusten ja kuvioiden luonti. Tässä vaiheessa malliin lisätään materiaalit, tekstuurit sekä värit. Jos materiaalin on tarkoitus näyttää muovilta, tulee siihen lisätä heijastava ja kiiltävä pinta. Mikäli sen on tarkoitus imitoida realistista lasia, tulee sen taittaa valoa kuin oikea lasi. Tekstuurit ja värit lisätään joko liittämällä jokaiseen pintaan oma kuva, tai suoraan maalaamalla mallin päälle. (Slick [Viitattu 16.2.2015].)

**Valotus:** Jotta 3D-mallit tulisivat näkyviin, tulee piirtoalustalle asetella digitaaliset valot. Valoilla on suuri merkitys siihen, millaisen tunnelman kappale herättää katsojissa. Kunnon valotuksella kappale näyttää realistiselta, mutta silti tekijä pysyy luomaan dramaattisuutta ja tunnelmaa. Myös varjot tulevat tärkeäksi osaksi tässä vaiheessa. (Slick [Viitattu 16.2.2015].)

**Animointi:** Kappaleen voi myös herättää eloon antamalla sille tapahtumasarjan. Animaatiossa esimerkiksi voi laittaa kappaleen kasvamaan, kutistumaan, liikkumaan, lentämään tai vaikka vaihtamaan kokonaan muotoa tai väriä. 3D-animointi on hyvin lähellä stop-motion-tekniikkaa. Yleisin tapa on antaa objektille ”luuranko”. Tällöin objektille voi antaa niveliä, joilla voi liikutella osia objektista ilman, että koko kappale liikkuu. Sitten määritellään animaation pituus, esimerkiksi 1000 kuvaa, sekä montako kuvaa näytetään sekunnissa (fps). Yleisimmät käytetyt renderöintitaajuuudet ovat 24, 30 ja 60 kuvaa sekunnissa. Tämän jälkeen kappaletta voi joko manuaalisesti liikuttaa joka kuvan kohdalla eri asentoon tai luoda tapahtumasar-

jan, jossa objekti muuttuu tietyn ajan kuluessa tietyn laiseksi. (Slick [Viitattu 16.2.2015].)

**Renderöinti ja jälkituotanto:** Viimeisin vaihe mallinnuksessa on renderöinti, joka muuntaa 3D-näkymän 2D-kuvaksi tai animaatioksi. Tässä vaiheessa liitetään yleensä mukaan spesiaaleffektejä, kuten animaatioissa sumu, savu ja räjähdykset. Myös manipulointi onnistuu renderöinnin jälkeen, jos on tarvetta hallita kirkkautta, kontrastia tai värejä. Nämä tehdään kuvanmuokkausohjelmilla. (Slick [Viitattu 16.2.2015].)



Kuvio 3. Grafiikkaputken vaiheet (The Pipeline Behind Modeling and Animating a Game Character in Zelgor 2013.)

### 3.3 3D-grafiikka internetselaimessa

3D-grafiikka tuli ensimmäisen kerran käyttäjien internetselaimiin vuonna 1997, kun julkaistiin VRML (Virtual Reality Modeling Language) ISO-standardi, joka teki mahdolliseksi kolmiulotteisen interaktiivisen vektorigrafiikan luonnin tekstoreineen näytettäväksi VRML:ää tukevissa selaimissa. Myöhemmin VRML-standardin kehitys jatkui X3D-nimisenä projektina. Vuonna 2005 julkaistiin XML-pohjainen X3D-standardin spesifikaatio. (Brutzman 1998.)

X3D oli julkaistessaan huomattavasti kehittyneempi teknologia 3D-grafiikan tuottamiselle. Koska X3D oli XML-pohjainen, mahdollisti se sovelluksien laajentamisen monipuolisemmiksi. Ulkoisia lähteitä ja tietokantayhteyksiä voitiin nyt ottaa käyttöön X3D-koodin tuottamisessa. (X3D 2001.)

Tästä syntyi suuri into kehittäjäyhteisöissä ja mainostajissa tuoda omia luonnoksia selaimiin muiden nähtäväksi (kuvio 4). Seurauksena syntyi muutamia uusia teknikoita 2000-luvulla kuten Java 3D sekä Flash. Tämä synnytti uuden ongelman, mikä oli käyttäjien laitteiden suorituskyky. 3D-grafiikan suosion leviämisen esteenä olivat vielä laskentatehollisesti heikot tietokoneiden prosessorit, muistit sekä pienikaistaiset tietoliikenneyhteydet. (The Evolution of the Web 2012.)



Kuvio 4. Vanha OpenGL-teknologialla toteutettu peli (Youtube 2015.)

Suurin käännekohta selaingrafiikalle syntyi vasta jonkin aikaa sitten, kun suurimmat selainvalmistajat toivat tuen grafiikkapiiriin kiihdytykselle selaimiinsa, mikä salli näytönohjaimen tehon käytön web-sovelluksissa. Suurimmat selainvalmistajat ovat Microsoft selaimellaan Internet Explorer, Google selaimellaan Google Chrome, sekä Mozilla selaimellaan Firefox. Tähän asti 3D:n mahdollistavat menetelmät edellyttivät erillisten lisäosien tai pluginien käyttöönottamista. Nykyisin käyttäjän on monesti asennettava jokin erillinen lisäosa pelkän sisällön näyttämisen vuoksi. Lisäosat sisältävät usein niin sanottuja "bloatwareja" ja vaativat usein paljon oikeuksia toimiakseen täydellisesti. (The Evolution of the Web 2012.)

Selainympäristössä 3D-tekniikoiden kehitys etenee hyvin nopealla tahdilla. Näytönohjaimen käytön mahdollistaminen selainympäristössä on edesauttanut monien uusien rajapintojen ja kehitysympäristöjen syntyä. Adobe toi uusimman Flash 11 - version julkistamisen myötä käyttäjille Stage3D:ksi kutsutun kehitysympäristön. Microsoftilla on myös oma 3D-työkalu nimeltä Silverlight. Suurin kiinnostus kehittäjillä on kuitenkin noussut uudehkoa WebGL-tekniikkaa kohtaan. Molemmat Stage3D ja WebGL sisältävät omat rajapinnat laitteistokiihdytetyn grafiikan tuottamiselle. Tässä työssä käytännön osuus toteutettiin WebGL-tekniikalla. WebGL on nykyisin paras tapa tuoda laitteistokiihdytettyä grafiikkaa esteettömästi käyttäjän internetselaimeen. Laitteistokiihdytetyn grafiikan pääsy esteettömästi käyttäjän internetselaimeen tarkoittaa poistunutta tarvetta asentaa tietokoneelle ylimääräisiä lisäosia. (The Evolution of the Web 2012.)

## 4 VIRTUAALIYMPÄRISTÖJEN VÄLINEITÄ

### 4.1 Virtuaalitodellisuuslasit

Oculus VR on amerikkalainen virtuaalitodellisuusteknologian yritys ja se perustettiin vuonna 2012, kun yhtiön perustaja Palmer Luckey sai idean kehittää virtuaalitodellisuuslasit, jotka olisivat paremmat sekä halvemmat mitä tuolla hetkellä oli tarjoilla pelaajille. Oculus Riftin prototyyppiä esiteltiin E3-messuilla kesäkuussa 2012. (Kickstarter 2012.)

Oculus Rift -virtuaalitodellisuuslaseissa on näyttö sisällä, joka peittää koko näkökentän digitaalisella kuvalla (kuvio 5). Kääntämällä päätä kuva liikkuu myös simulaatiossa mukana reaaliajassa, joka luo tunteen kuin olisi itse sisällä pelissä. Molemmat silmät katsovat eri linssin kautta ja näkevät eri osat näytöstä, mikä luo stereoskooppisen 3D-efektin. Molemmille silmille näytetään omaa digitaalista kuvaa hieman eri kulmasta kuvattuna imitoiden silmien toimintaa ja luoden näin syvyyden peliin. Katsesuunnan tunnistuksen ja syvyyden luomisen lisäksi Rift pystyy myös tunnistamaan liikkeitä koordinaattiakselilla infrapunatunnistuksen avulla. Jos esimerkiksi nojaa eteenpäin oikeassa elämässä, matkii peli tätä efektiä tekemällä samoin virtuaalimaailmassa. (Grubb 2014.)

Oheessa Oculus Rift DK 2 -version spesifikaatiot:

- resoluutio: 960 x 1080 pikseliä per silmä
- näyttöjen päivitystaajuus: 75 Hz, 72 Hz ja 60 Hz
- päivitysviive: 2 ms, 3 ms
- näkyvyyskulma: 100°
- liittimet: USB 2.0 ja HDMI 1.4b
- sensorit: Gyroskooppi, kiihtyvyysanturi, magneettianturi
- sensoreiden päivitystaajuus: 1000 Hz
- paikkatunnistus-sensori: Lähi-infrapuna CMOS sensori, 60 Hz päivitystaajuus
- hinta: 300–350e. (Grubb 2014.)

Työn käytännön osuudessa käytettävät virtuaalitodellisuuslasit ovat Oculus Rift DK 2. Työn tekohetkellä ei ollut markkinoilla muita tähän työhön soveltuvia virtuaalitodellisuuslaseja. Kuluttajille suunnattu versio virtuaalilaseista on odotettavissa julkaistavaksi jossain vaiheessa vuotta 2015, mutta siitä on saatavilla kehittäjille suunnatut Oculus Rift Development Kit 1 & 2 -versiot.



Kuvio 5. Oculus Rift avattuna (iFixit 2014.)

## 4.2 3D-kamera

Nimble VR perustettiin vuonna 2012. Yhtiö kehitti aluksi käsien tunnistusteknologiaa, joka toisi paremman käyttökokemuksen moneen asiaan. 3D-kamerateknologian sekä yhtiön henkilöstön kehittymisen myötä yhtiö kykeni ylitse pääsemään esteet, jotka pitivät liikkeentunnistusta hankalana toteuttaa. (Starr 2014.)

Yhtiö aloitti työskentelyn värihanskojen parissa, josta se pian siirtyikin liikkeen paikantamisen kehittelyyn usean Kinect-kameran avulla, johtaen lopulta siihen, että yhtiö kehitti oman pienen 3D-kameran, joka nimettiin Nimble Senseksi. Nimble Sense on tarkoitettu liitettäväksi kiinni Oculus Riftin etupaneeliin (kuvio 6). Elokuussa 2014 yhtiö loi Kickstarter-projektin, jonka tavoitteena oli esitellä tehokkaan luurankomallisen kädentunnistusmenetelmän, joka toisi käsien liikkeet mukaan virtuaalitodellisuuteen. (Starr 2014.)

Nykyään yhtiö on liittynyt yhteen Oculus-tiimin kanssa ja kehittävät yhteistyössä entistä tarkempaa ja kompaktimpaa käsientunnistusteknologiaa osaksi virtuaalitodellisuutta. (Starr 2014.)

Tässä työssä ei ollut käytössä Nimble VR:ää, mutta tulevaisuutta ajatellen voisi tämä teknologia tuoda monia etuja kyseiseen käyttötarkoitukseen.



Kuvio 6. Nimble VR kiinnitettynä Oculus Riftissä (Gilbert 2014.)



### 4.3 Peliohjain

Virtuix Omni on virtuaalimaailma käyttöön tarkoitettu peliohjain, joka mahdollistaisi fyysisen liikkeen siirrettäväksi virtuaalimaailman liikkeeksi. Tuote on tavallaan kuin liukas kovera alusta, jonka päällä voi kevyesti liikkua mihin tahansa suuntaan (kuvio 7). Laitetta on tarkoitettu käytettäväksi mukana tulevilla matalakitkakengillä. Liikkeet juoksumatolta siirtyvät virtuaalimaailmaan anturien kautta. Laitteessa on myös vyötärön ympärille tuleva tukirengas, jonka tarkoitus on tukea pelaajaa tiukasti paikoilleen. (Virtuix 2014.)

Virtuix Omni on myös Kickstarter-projekti, joka sai 1.1 miljoonaa dollaria rahoitusta kampanjan aikana. Myöhemmin yhtiö sai 8 miljoonaa dollaria lisärahoitusta. Virtuix Omnia on tarkoitettu käytettäväksi virtuaalilasien, esimerkiksi Oculus Riftin kanssa, luomaan täydellisen uppoutuvuuden ja läsnäolon tunteen. (Virtuix 2014.)

Myöskään Virtuix Omnia ei ollut käytössä tässä opinnäytetyössä, mutta tämäkin varmasti toisi lisää realismia tunnetta sovelluksiin (Virtuix 2014).



Kuvio 7. Virtuix Omni -laite (Digitalized technologies [Viitattu 5.3.2015].)

## 5 KÄYTETYT OHJELMAT JA TEKNOLOGIAT

### 5.1 Web-ohjelmointi

Tässä opinnäytetyössä käytännön osuudessa tutkittava MyHome-sovellus on luotu useamman eri web-ohjelmointikielien yhteisenä tuloksena. Myös työn tekijän tekemä osuus painottuu suuresti web-ohjelmointiin.

#### 5.1.1 HTML5

HTML5 on HTML (Hypertext Markup Language) -merkkaukielen uusin versio. Sen on kehittänyt yhteistyössä Apple, Mozilla, Opera sekä W3C. Lokakuussa 2014, HTML5-standardi todettiin valmiiksi. HTML5 on periytynyt XML (Extensive Markup Language) -merkkaukielestä ja siitä on periytynyt logiikka, että kaikki data tulee tagien sisään. Suurin ero näissä kahdessa on se, että XML on tarkoitettu kuvaamaan rakennetta ja web-dataa, kun taas HTML on tarkoitettu näyttämään visuaalisesti dataa. HTML5 on nykyaikana eniten käytetty teknologia datan rakentamiseen ja näyttämiseen internetissä. HTML5 on tarkoitettu helposti luettavaksi ihmiselle, mutta silti täysin yhteensopivaksi tietokoneille ja laitteille. HTML5 on luotu sisältämään HTML 4:n lisäksi myös XHTML 1 (Extensive Hypertext Markup Language version 1) sekä DOM Level 2 (Document Object Model Level 2) HTML-teknologioiden tuomat edut. (W3C 2012.)

HTML5 tukee myös erittäin hyvin mobiililaitteita sekä sulautettuja järjestelmiä. Tarkoituksena on myös, että HTML5 olisi hyvin kevyt, jotta pienitehoisetkin laitteet voisivat hyödyntää sitä. (W3C 2012.)

HTML5 erottuu muista, koska se tukee hyvin laajasti eri medioiden tuonnin ja näyttämisen selaimessa. Uusina elementteinä HTML5 tukee videoiden <video>, äänien <audio>, 3D-objektien <object>, sekä piirtoalueiden <canvas> tuonnin selaimen ilman ylimääräisiä rajapintoja tai lisäosia. Varsinkin <canvas> on hyvin suuri askel eteenpäin, sillä sen avulla voi esittää hyvin monipuolisia demoja. WebGL mahdollistaa jopa melko raskaiden graafisten elementtien suorittamista

selaimes, hyödyntäen laitteen grafiikkapiiriä. Muuta mainittavan arvoista on, että HTML5 sisältää paikannuksen (GeoLocation), paikallisen tiedonsäilönnän sekä mahdollisuuden ajaa sovelluksia ilman internetyhteyttä. Paikannus on varsinkin hyödyllinen, jos haluaa ohjata käyttäjät ulkomaan osoitteista eri kielille käännetuille verkkosivuille. Tiedonsäilöntä taas mahdollistaa esimerkiksi peleissä tallennusten tekemisen selaimeen käyttämällä evästeitä. Sovelluksien ajo ilman internetyhteyttä mahdollistaa pelien tai muun grafiikan suorituksen omalta koneelta selaimessa (Makzan 2011, 8-11.)

### 5.1.2 CSS 3

CSS 3 (Cascading Style Sheets 3) on tyyliohje, jota käytetään merkintäkielellä, esimerkiksi HTML- tai XML-kirjotetun dokumentin ulkoasun ja formaatin muotoiluun. CSS:ää on mahdollista käyttää minkä tahansa XML-pohjaisen merkintäkielen kanssa. Tyyliä on mahdollista määritellä suoraan HTML-dokumentin sisällä tai erillisessä tiedostossa, jota kutsutaan HTML-tiedostosta. (Mozilla Developer Network 2014.)

CSS 3 on jaettu moduuleihin eli ominaisuudet eivät ole riippuvaisia toisistaan. Moduulit ovat tarkoitettu käytettäväksi itsenäisesti. Tämä tarkoittaa sitä, että toisia ominaisuuksia voi määritellä ja suorittaa eteenpäin riippumatta siitä, onko aiemmat ominaisuudet suoritettu. CSS 3 tukee monipuolisesti nykyaikaisia multimediodien muokkausominaisuuksia. Esimerkiksi transform (muodonmuutos) ja animation (animaation) ovat monipuolisia tapoja muotoilla ja editoida elementtejä. Näillä on mahdollista esimerkiksi muuttaa elementin väriä, kokoa, paikkaa, muotoa, tuoda esiin tai hävittää koko elementti. Suurimmat selaimet tukevat näitä tyyliä, mutta jokainen vaatii oman etuliitteensä, esimerkiksi Operalle "-o-" sekä Firefoxille "-moz"-etuliite. (Mozilla Developer Network 2014.)

### 5.1.3 Javascript

Javascript-ohjelmointikieli on tämän opinnäytetyön kannalta oleellisin teknologia. Käytännön osuudessa tutkittava MyHome-sovellus on toteutettu suurimmaksi osaksi Javascriptillä. Myös sovelluksen tuonti Oculus Riftiin tulee tapahtumaan Javascriptiä ja sen apukirjastoja hyödyntäen.

Javascript on hyvin monipuolinen ohjelmointikieli, joka toimii yhteistyössä selaimen kanssa. Koodirivit suoritetaan selaimessa siinä järjestyksessä kun ne tulevat vastaan. Javascript-kielellä on mahdollista muuttaa staattinenkin verkkosivu interaktiiviseksi. Javascriptillä on mahdollista tunnistaa käyttäjän kellonaika ja luoda sen perusteella toimintoja, esimerkiksi tervehdys ”Hyvää huomenta”, mikäli käyttäjän aikavyöhykkeellä on aamu. Ongelma tässäkin web-kielessä on se, että julkaistua koodia on hyvin hankala piilottaa, tehden sen erittäin helpoksi varastaa ilman suurta tuntemusta selainten käyttöön. (w3schools [Viitattu 17.2.2015].)

Javascriptillä on aikojen saatossa ollut useampi nimi, muun muassa Mocha, LiveScript, JScript ja ECMAScript. Javascript on yksi maailman suosituimmista ohjelmointikielistä. Virtuaalisesti ainakin jokaisella maailman tietokoneella on yksi Javascriptiä hyödyntävä sovellus asennettuna ja se on aktiivisessa käytössä. Javascriptin suosio perustuu täysin sen rooliin maailmanlaajuisen tietoverkon (World Wide Web) skriptikielenä. Vaikka kielen etuliitteessä on sana ”Java”, ei sillä siltikään ole mitään tekemistä Oraclen kehittämän Java-ohjelmointikielen kanssa. Javascript ei ole oikea ohjelmointikieli, vaan skriptikieli. Kuitenkin Javascriptillä voi saada aikaan melkein saman tuotoksen kuin C-ohjelmointikielellä. Näissä kahdessa syntaksikin on hyvin pitkälti sama. C-ohjelmointikieleen verrattuna Javascript vaihtaa suorituskykyä ilmaisuvoimaan sekä dynaamisuuteen. Javascript tiedostot ovat ”.js”-päätteisiä. Javascript suunniteltiin Netscape Navigator-selaimeen. Sen suosio kuitenkin teki siitä standardin melkein kaikkiin selaimiin. Javascript sopii myös hyvin suureen osaan ei-web-pohjaisiin sovelluksiin. (Crockford 2001.)

Javascript-koodauksen helpottamiseksi ja nopeuttamiseksi on kehitelty runsaasti erittäin käytännöllisiä, aikaa säästäviä, käyttöjärjestelmästä riippumattomia, funktioita sisältäviä apukirjastoja, jotka liittämällä omaan dokumenttiin voi luoda nopeasti ja yksinkertaisesti hankalia, paljon koodaamista ja tietämystä vaativia toimintoja.

Suuri osa näistä kirjastoista on yksityisten henkilöiden luomia. Tässä opinnäytetyössä tärkeimmät kirjastot, joita käytetään, ovat Three.js sekä VRRenderer.js, näistä kerrotaan myöhemmin. Javascriptin päälle on myös rakennettu erilaisia runkoja koodaamisen monipuolistamiseksi ja nopeuttamiseksi. Yksi mainittava näistä on jQuery, joka on nopea, pieni, sekä toimintorikas Javascript-kirjasto. JQuery muuttaa hyvin pitkälti koko koodin syntaksia lyhyemmäksi ja kompaktimmaksi. (jQuery [Viitattu 17.2.2015].)

Toinen hyvin paljon suosiota saanut runko on Node.js. Tämän avulla voi tuoda palvelinteknologian selaimen yhteyteen. Tämä edesauttaa suuresti reaaliaikaisten web-sovellusten kehittämistä ja toteuttamista. (Winn 2013.)

#### **5.1.4 PHP**

MyHome-sovellus toteutettiin aluksi HTML5 kielellä, mutta muunnettiin myöhemmin PHP-muotoiseksi, jottei koko sovellusta saisi kopioitua suoraan selaimesta. Sovelluksessa käytettiin myös PHP:tä asiakaspalautetoiminnon luonnissa.

PHP (PHP: Hypertext Preprocessor) on laajasti käytetty avoin yleiskäyttöinen skriptikieli, se on tarkoitettu etenkin web-kehitykseen. PHP-koodi voidaan upottaa suoraan HTML-koodin sekaan. Suurin ero PHP ja Javascriptin välillä on se, että PHP on täysin palvelinpuolen koodia. Se siis ajetaan palvelimella, ja palvelin palauttaa vastauksen HTML-muodossa, joka näytetään käyttäjälle selaimessa. PHP onkin hyvä tapa piilottaa koodia, sillä se ei näy loppukäyttäjän selaimessa. PHP:n mainitaan olevan hyvin yksinkertainen ja helppo kieli uusille tulokkaille, mutta se tarjoaa myös kehittyneempiä toimintoja ammattilaisille. (php [Viitattu 17.2.2015].)

Facebook on tunnetuimpia verkkopalveluita, joka on luotu vahvasti PHP:hen luottaen. Yksi yleisimmistä käyttökohteista PHP:llä on sen sisäänrakennetut MySQL (My Structured Query Language) -tietokantafunktiot. (php [Viitattu 17.2.2015].)

## 5.2 Teknologiat

Tässä työssä tutkittavan MyHome-sovelluksen kannalta uudet teknologiat ovat välttämättömiä. Pelkällä web-ohjelmoinnilla ei voisi synnyttää raskaita graafisia ympäristöjä web-maailmaan käyttäjille näytettäväksi. Tässä avuksi tulivat erilaiset teknologiat, joita nykyään on integroituna suurimmissa selaimissa.

### 5.2.1 OpenGL

Vuonna 1990 alkunsa saanut OpenGL (Open Graphics Library) on Silicon Graphics Inc:n (SGI) luoma rajapinta, joka perustuu saman yhtiön luomaan IRIS GL:ään, joka on 3D-ohjelmointirajapinta SGI:n IRIS-työasemille. IRIS GL ei kuitenkaan ollut siirrettävissä muihin laiteympäristöihin, ja SGI:n suunnitelmissa olikin parantaa IRIS GL:n siirrettävyyttä. Tätä varten muodostettiin useita, suurista yrityksistä (mm. Apple ja Microsoft) koostuvia työryhmiä edesauttamaan OpenGL-rajapinnan kehitystä. Vuonna 1992 julkaistiin ensimmäinen OpenGL 1.0 -rajapinta, ja vuonna 2000 OpenGL:n lähdekoodi jaettiin julkiseen levitykseen. (OpenGL 2015.)

3D-rajapinnan tehtävä on ottaa sovellukselta vastaan standardinmukaista tietoa siitä, mitä sovellus haluaa näytölle piirrettävän ja muuttaa se sitten laitteistolle ohjauksikäskyiksi, jotta saavutettaisiin ohjelman haluama toiminta. OpenGL mahdollistaa kaksi- ja kolmiulotteisten graafisten näkymien piirtämisen. OpenGL on sekä standardisoitu ohjelmointirajapinta että rajapinnan toteutus. Yksinkertaisesti sanoen se on joukko funktioita, joilla on laiteriippumattomasti sama syntaksi, ja niiden voidaan odottaa toimivan samalla tavalla jokaisella OpenGL-rajapintaa tukevalla laitteistolla. OpenGL on niin kutsuttu ”välittömän piirtämisen rajapinta”, jossa malli piirretään samanaikaisesti, kun rajapinnalle syötetään geometrisiä objekteja piirrettäväksi. (OpenGL 2015.)

OpenGL on nimensä mukaisesti avoin grafiikkakirjasto sekä ohjelmointirajapinta. OpenGL on hyvin suosittu videopeleissä ja CAD-ohjelmistoissa sen alustariippumattomuuden ja ohjelmointikielen vuoksi. OpenGL mahdollistaa tietokoneen grafiikkaohjaimen hyödyntämiseen 2D- ja 3D-grafiikan piirtämisessä. (OpenGL 2015.)

OpenGL valmistui ja siitä tuli standardi ARB:n (Architectural Review Board) toimesta vuonna 1992. ARB:n perustajajäseniin kuuluivat SGI, Microsoft, Intel, DEC ja IBM. OpenGL-rajapinnan kehitys on sen jälkeen pitkälti perustunut näytönohjainvalmistajien ja esimerkiksi pelinkehittäjien luomiin OpenGL-lisäosiin ja vaatimuksiin. OpenGL-rajapinnasta harvoin ilmestyy uusia versioita, mutta näytönohjainvalmistajat voivat itse lisäosillaan tuoda uudet ominaisuudet rajapinnalle. (OpenGL 2015.)

### 5.2.2 Canvas-elementti

Canvas on HTML5 mukanaan tuoma elementti ja rajapinta (API). Canvas-elementti tarjoaa skriptejä resoluutioriippuvaiselle bitmap-ikkunalle, joita voidaan käyttää grafiikoiden renderöintiin. Canvasta käyttämällä on mahdollista renderöidä peligrafiikoita, taidetta tai muita visuaalisia kuvia suoraan lennosta hyödyntäen Javascriptiä. (WHATWG [Viitattu 17.2.2015].)

Apple kehitti Quartz 2D -nimisen kaksiulotteisen grafiikan renderöinti rajapinnan Mac OS X -käyttöjärjestelmille 2000-luvun alussa. Vuosien kehittyessä myös toiminnot monipuolistuivat. Lopulta siitä tuli hyvin toimivaksi todettu ratkaisu, joten Mozilla ja Opera halusivat sen selaimiinsa. Canvas kehiteltiin pitkälti Applen Quartz 2D:n pohjalta ja se onkin oikeastaan vastike muille alustoille Quartz 2D:stä. (Sitepoint 2014.)

Alkuperäinen Canvasin piirtorajapinta sai myöhemmin seurakseen WebGL-rajapinnan. Alkuperäistä piirtorajapintaa on mahdollista käyttää kutsumalla sitä "2D" -kontekstilla. Uutta HTML5 WebGL -rajapintaa voidaan kutsua "webgl" -kontekstilla. Suurin ero näiden kahden välillä on se, että WebGL-rajapinta hyödyntää tietokoneen näytönohjainta grafiikan piirtämisessä. (Sukan 2009.)

### 5.2.3 AJAX

AJAX (Asynchronous Javascript and XML) on uudehko tekniikka luoda parempia, nopeampia ja entistä interaktiivisempia web-sovelluksia käyttäen apuna HTML:ää, XML:ää, Javascriptiä ja CSS:ää. AJAX käyttää XHTML-tekniikkaa sisältöön, CSS:ää muotoiluun, sekä DOM:ia ja Javascriptiä dynaamisen sisällön näyttämiseen. Tavanomaiset web-sovellukset siirtävät tietoa palvelimen ja käyttäjän välillä synkronisilla pyynnöillä. Toisin sanoen tämä tarkoittaa esimerkiksi sitä, että kun täytetään lomake ja painetaan "lähetä"-nappia, niin sen jälkeen käyttäjä ohjataan uudelle sivulle, jossa lukee palautetta palvelimelta, esimerkiksi "Kiitos vastauksesta!". (Ajax [Viitattu 17.2.2015].)

AJAX mahdollistaa epäsynkroonisen tiedonsiirron, mahdollistaen tiedon haun ja esittämisen palvelimelta kesken sovelluksen ajona ilman, että välissä täytyy vaihtaa tai päivittää sivua. XML:ää käytetään useimmiten formaattina, jona palvelimelta saatu vastaus tulkitaan. AJAX on selainteknologia ja täysin riippumaton web-palvelimen ohjelmistosta. Kuitenkin AJAX on täysin riippuvainen Javascriptistä. (Ajax [Viitattu 17.2.2015].)

### 5.2.4 Websocket

HTML5 mukana tuli uusi Websocket-tekniikka korvaamaan vanhemman AJAX-tekniikan. Kumpikin tekniikka perustuu siihen, että ne hakevat palvelimelta tietoa ilman, että selaimen täytyy päivittää sivua haun toteuttamiseksi. AJAX käyttää tapaa, jossa luodaan palvelimelle uusi yhteys, ladataan palvelimelta haluttu data ja vastaanotetaan tiedot. Tämä tehdään aina kun ohjelman käyttäjä haluaa päivittää tietoja. Websocket luotiin yksinkertaistamaan prosessia ja mahdollistamaan nopeammat yhteydet. (Selim 2011.)

HTTP (Hypertext Transfer Protocol) -protokollan tavoin Websocket käyttää kaksisuuntaista yhteyttä palvelimeen. Turvallisuusmalli tekniikalle on sama, mitä tavallisiin nettisivujen lataamispyyntöihin muutenkin käytetään. Tällä tavoin viestien kulku ei pysähdy palvelinten palomuuereihin, tuoden huomattavasti lisää nopeutta. Tekniikkaan kuuluu myös yhteyskokeilu, joten asiakasohjelman ei tarvitse huoleh-



tia yhteyden säilymisestä. Yhteyskokeilun hoitaa selain ja palvelimelle luotu ohjelma. Websocket-tekniikka on verrattavissa peleissä käytettäviin palvelimiin, mutta Websocketin käyttämä portti on jatkuvasti käytettävissä useilla palvelimilla. (Melnikov 2011.)

Etuna Websocketissa on se, että palvelimen ja Websocketin välinen työskentely on kevyttä. Websocketin toiminta perustuu siihen, että se käyttää vain yhtä yhteyttä palvelimeen, suorittaa palvelimelta valitun ohjelman kerran ja alkaa vastaanottamaan tietoa. AJAX-tekniikka puolestaan tekee operaation uudelleen jokaisella tietopyyntökerralla. Websocket palvelin toimittaa myös uusimman tiedon selaimelle automaattisesti. Websocket tuo palvelintekniikan HTTP-ympäristöön, joka mahdollistaa nopean tiedonsiirron. Tätä on mahdollista hyödyntää esimerkiksi moninpelien yhteyksissä. Moninpelaaminen selaimessa ei ole uusi asia, mutta Websocket aukaisi portit yhä kehittyneempään sovelluskehitykseen selainympäristössä. (WebSocket 2013.)

MyHome-sovelluksen luontivaiheessa Websocket-tekniikkaa ei ollut vielä tehokkaasti saatavilla, vaan tilalla käytettiin vanhempaa AJAX-tekniikkaa. Kuitenkin lähitulevaisuudessa tullaan varmasti Websocket ottamaan käyttöön sovelluksessa, mahdollistaen esimerkiksi monikäyttäjän tuen ja tehokkaamman asiakaspalautteen luomisen.

### 5.2.5 WebGL

WebGL on alustariippumaton, rojaltivapaa rajapinta, jota käytetään 3D-grafiikan renderöintiin selaimessa. WebGL on luotu OpenGL ES 2.0 (Embedded System 2.0) pohjilta ja käyttää OpenGL:n varjostinkieltä, GLSL:ää (OpenGL Shading Language), sekä tarjoaa OpenGL-rajapinnasta tutun lähestymistavan. WebGL toimii HTML5 Canvas-elementissä ja sillä on täysi integraatio DOM:n ulkoasuun. WebGL on siis DOM-rajapinta, joka tarkoittaa sitä, että sitä voi käyttää kaikilla DOM-yhteensopivilla kielillä, yleisin on kuitenkin Javascript. Suurimmat selainvalmistajat kuten Google, Opera, Mozilla ja Apple kuuluvatkin Khronos-konsortion WebGL-kehitysryhmään monen muun 3D-grafiikkakehittäjän kanssa. (WebGL 2011.)

Koska WebGL pohjautuu pitkälti OpenGL-tekniikkaan ja se integroidaan suosituimpiin selaimiin, tarjoaa se monia etuja kuten:

- tukee ja hyväksyy suosituimpia 3D-grafiikan standardeja
- selainriippumaton ja käyttöjärjestelmäriippumaton
- erittäin tiukasti integroitu HTML-sisältöön, mukaan lukien kerrosten sommittelu, vuorovaikutus muiden HTML-elementtien kanssa, sekä käyttäjän syötteen kuuntelu ja toimintojen suoritus syötteestä riippuen
- laitteistokiihdytetty 3D-grafiikka selainympäristöön
- skriptausympäristö, joka tekee helpoksi 3D-grafiikkaprototyyppien kokeilun, koska on mahdollista suoraan renderöidä ja virhekorjata eli debugata ohjelmaa.(WebGL 2011.)

WebGL on uusi tekniikka, mutta sillä on kilpailijoita, jotka ovat olleet markkinoilla jo pidempään. Kuitenkin web-kehittäjä Florian Boesch (2013) on tuonut ilmi hyviä syitä, miksi tulisi käyttää juuri WebGL:ää. Florian mainitsee esimerkiksi, että kilpailijoihin nähden WebGL-rajapinnan kutsumiseen tarvitaan vähemmän koodia. Ei ole myöskään tarvetta luoda tiedostoja, lisätä kirjastoja tai kutsua erikseen apuvälineitä. (Boesch 2013.)

WebGL-rajapinnan kutsumiseen tarvitaan vain kolme riviä: Canvas-elementin luonti, sen sijoittaminen koodiin, sekä kontekstin määrittely WebGL-rajapinnaksi. Selain tekee suuren osan töistä ohjelmoijan puolesta, mitä pitäisi muuten tehdä toisessa ympäristössä. (Boesch 2013.)

Grafiikkaohjelmoinnissa näytönohjaimien ajureiden tuki ja kehittyminen on hyvin suuressa roolissa toteutettavaan lopputulokseen nähden (kuvio 8). WebGL on kuitenkin yksi tuetuimmista rajapinnoista nykyään. Tämä johtuu pitkälti siitä, että sen testaaminen erilaisilla kokoonpanoilla on nopeaa ja helppoa. Testaaminen onnistuu helposti laittamalla vain testisivun osoite selaimen osoiteriville ja valitsemalla testin aloittamisen. Joissakin muissa ympäristöissä tämä saattaisi vaatia erillisten ohjelmien ja ajureiden lataamista. (Boesch 2013.)

OpenGL-rajapinnan tavoin WebGL:n on mahdollista laajentaa ja lisätä toimintoja ilman suuria versiopäivityksiä. WebGL erottuu muista web-ympäristön 3D-teknologioista siinä, että se voi hyödyntää uusia ominaisuuksia saman tien. Laajennukset tulevat pääasiassa Khronos Groupin kautta, mutta kehittäjät voivat myös itse luoda laajennuksia yhteisölle. (Boesch 2013.)



Kuvio 8. WebGL:llä toteutettua selaingrafiikkaa (Tarun 2013.)

### 5.3 Apuvälineet

Apuvälineet tarkoittavat tässä opinnäytetyössä lähinnä apukirjastoja tai sovelluksia, joita on tehty helpottamaan koodaajan ja testaajan työtä. Apukirjastoja on luotu monia erilaisia, eri tarkoituksiin. Apukirjastoja löytyy helposti esimerkiksi GitHub-sivustolta. Nämä voi ottaa käyttöön lisäämällä HTML-sivun alkuun `<script>` tagi ja liittämällä sen kautta kirjaston joko internetistä tai lokaalista osoitteesta.

### 5.3.1 Three.js

Three.js on apukirjasto grafiikan luomiseen WebGL-rajapintaa käyttäen. Normaalisti jonkun helpon muodon, esimerkiksi pallon luominen voisi vaatia tuhansia rivejä Javascript- ja varjostin-koodia, mutta tämän apukirjaston avulla siihen tarvitaan vain muutama rivi. (mrdoob 2013.)

Three.js sai alkunsa kun GitHub-palvelussa mrdoob-nimimerkillä esiintyvä henkilö alkoi vuonna 2010 kehittämään kyseistä kirjastoa, joka saavutti suuren suosion myöhemmin WebGL-tekniikan yleistyessä. Kirjasto toimii avoimen lähdekoodin periaatteella eli kehittämiseen voivat osallistua myös muutkin henkilöt. Tähän mennessä kehittämisessä onkin ollut osallisena yli 400 henkilöä. (mrdoob 2015.)

Kehittäjien kirjavuudesta johtuen dokumentaatio on hyvin vajavaista ja sen taso on vaihtelevaa. Joistakin toiminnoista voi olla hyvin vaikea löytää tietoa, mutta onneksi WebGL mahdollistaa helpon ja nopean kokeilun.

Three.js on kehitetty täysin grafiikan luomiseen 3D-objekteista varjostimiin ja erilaisiin tehosteisiin. Siihen on luotu myös pelintekoon keskittyvä lisäosa THREE.js. Tähän kuuluu esimerkiksi näppäimistön tai peliohjaimen kuuntelu. (THREE.js 2015.)

Three.js sisälsi aluksi myös törmäyksen tunnistuksen, mutta tämä poistettiin myöhemmin. Tilalle luotiin kuitenkin fysiikkakirjasto nimeltä Physi.js, joka on toteutettu Three.js-runkoa käyttäen, ja toi mukanaan tämän ja paljon muuta hyödyllistä 3D-objektien liikefysiikkaan liittyen. (Prall [Viitattu 17.2.2015].)

Three.js-kirjaston uudemmissa versioissa ei välttämättä toimi kaikki toiminnot, joita aiemmissa versioissa on ollut.

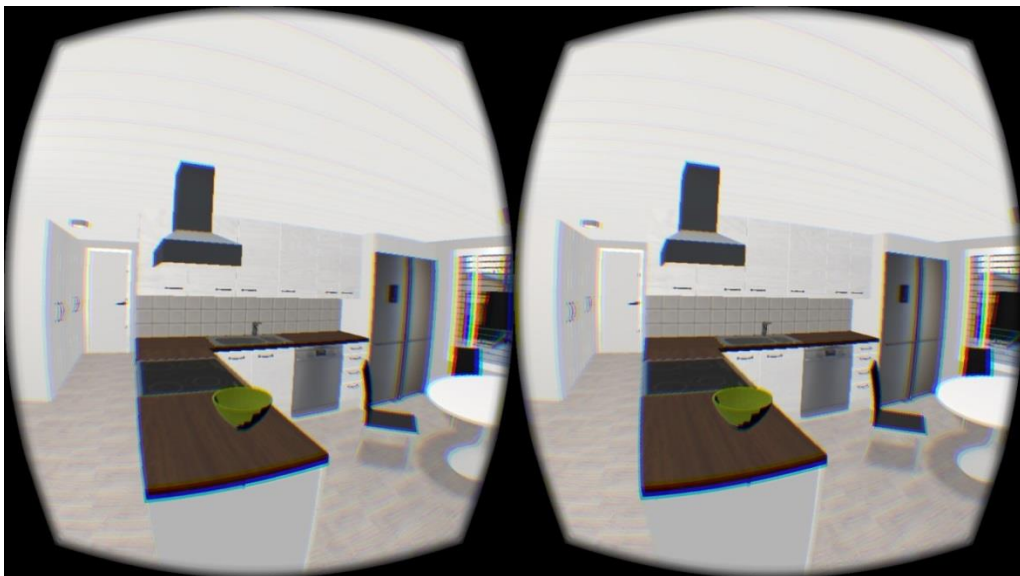
### 5.3.2 VRrender.js

VRrender.js on Rye Terrell -käyttäjän lisäämä apukirjasto GitHubissa, joka sallii WebVR-rajapinnan käytön suosituimmissa selaimissa. Tämä ei kuitenkaan ole tullut vielä osaksi julkisia selaimia. VRrender.js on luotu laskemaan näkökentät, luomaan omat kamerat molemmille silmille sekä muovaamaan kuva oikean-näköiseksi virtuaalilaseille. (Terrel 2014a.)

WebVR:n tehtävä on tunnistaa kytketty laite ja renderöidä kuva selaimen Canvas-elementistä virtuaalilasin näyttöön. Toinen tärkeä tehtävä on muuntaa lasien kääntely elementissä tapahtuvaksi näkökentän kääntelyksi (Jones 2014.).

### 5.3.3 VR Enabled Chrome

Tässä työssä käytettiin WebVR-enabled Nightly Chromium -selainta työn testauksessa. Nightly Chromium on Googlen Chromium -kehittäjäselaimesta tehty muunnos, joka sallii WebVR:n käytön (kuvio 9). Chromium on Chrome-selaimesta tehty muunnos, joka sisältää paljon kehittäjille suunnattuja työkaluja ja ominaisuuksia, kuten mobiiliemulaattorin sekä mahdollisuuden suorittaa riskialtista koodia. (Jones 2014.)



Kuvio 9. Laseissa näkyvä kuva imitoidaan tietokoneen näytölle.

#### **5.3.4 XAMPP**

Sovelluksen ajaminen selaimessa lokaalisti Windows-pohjaisilla laitteilla ei ole sallittua oletuksena, vaan se vaatii parametrien lisäämisen selaimen suorituskomentoon. Tämä saattaa kuitenkin aiheuttaa turvallisuusongelmia selatessa muita internetsivuja. Toinen vaihtoehto on luoda virtuaalinen lokaali palvelin, josta ajettuna sovellus saavat käyttöönsä tarvittavat resurssit. (Apache friends 2015.)

Tässä työssä käytettiin XAMPP-nimistä sovellusta, jolla luotiin virtuaalipalvelin työn tekijän tietokoneelle.

## 6 MYHOME

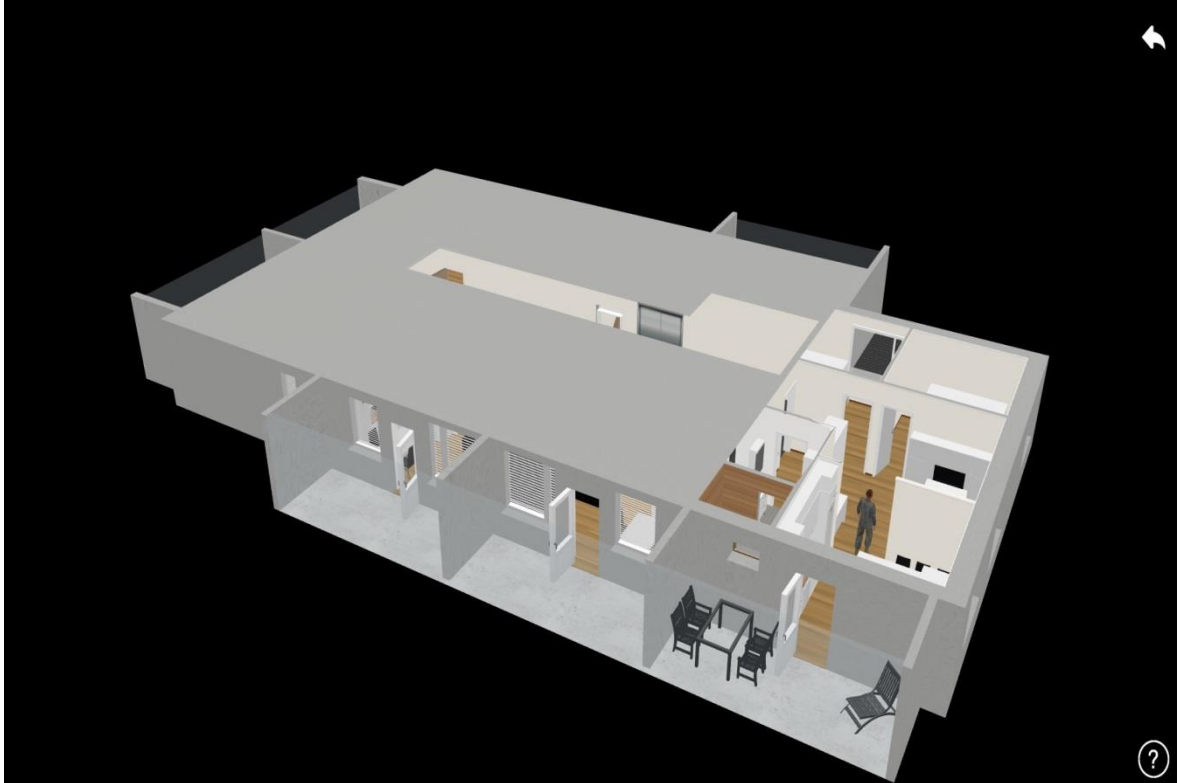
### 6.1 Näkymät

Tässä opinnäytetyössä integroitiin Oculus Rift-virtuaalilasit osaksi MyHome-sovellusta. Sovellus on interaktiivinen, kolmiulotteinen virtuaaliympäristö (kuvio 10), tarkoituksena on tarjota käyttäjille mahdollisuuden tutkia tulevia asuntoja etukäteen omalta tietokoneelta tai mobiililaitteelta. Se toimii kaikissa WebGL:ää tukevissa selaimissa. Sovelluksessa on hyödynnetty paljon pelimäisiä mekaniikoita. Kun sovellus käynnistetään, hahmo ”syntyy” valittuun huoneistoon. Huoneistossa hahmolla voi liikkua käyttäen näppäimistöä. Itse liikkuminen tapahtuu w-, a-, s- ja d-näppäimillä ja katselukulman kääntely tapahtuu nuolinäppäimillä.



Kuvio 10. MyHome 3D-näkymä.

Sovelluksessa on myös paljon lisätoimintoja näkymän muotoiluun ja muokkaukseen. On mahdollista esimerkiksi vaihtaa pintojen materiaaleja, piilottaa huonekalut, muotoilla väliseinän kokoa sekä vaihtaa näkymää. Näkymiä on kolme erilaista: 3D-näkymä, joka tarjoaa katselukulman hahmon silmien tasolta (kuvio 10).



Kuvio 11. Ylänäkymä.

Ylänäkymä, joka mahdollistaa huoneiston tutkimisen yläperspektiivistä (kuvio 11).

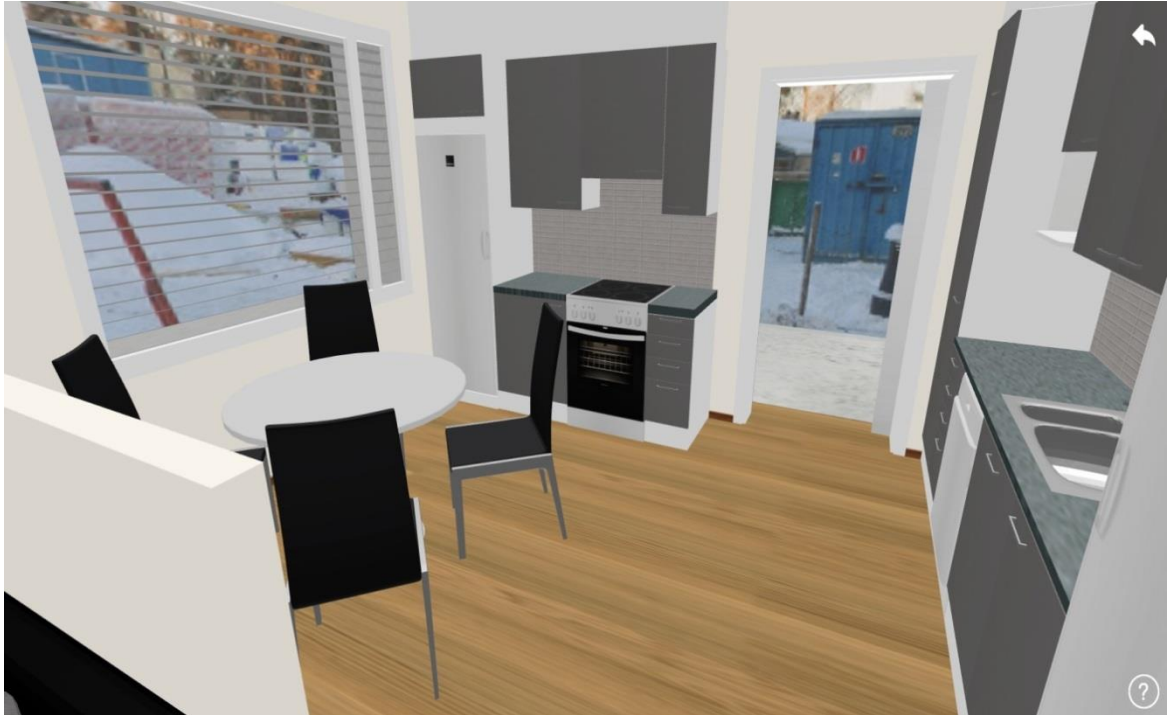


Viimeisenä on puolitetty näkymä, jossa molemmat aiemmat näkymät ovat jaettu samaan ruutuun, toinen yläpuolelle ja toinen alapuolelle näyttöä (kuvio 12). Puolitetussa näkymässä on se hyöty, että hahmoa voi siirtää haluttuun paikkaan ylänäkymässä klikkaamalla siihen kohtaan, mihin haluaa hahmon siirtyvän.



Kuvio 12. Puolitettu näkymä.

Pintojen materiaalien vaihto (kuvio 13) onnistuu osoittamalla haluttua pintaa, jolloin se alkaa hohtamaan, ja painamalla hiiren vasemmalla painikkeella. Väliseinän muotoiluun ja huonekalujen näyttöön liittyvät toiminnot löytyvät sivupaneelistä. Niihin on myös omat pikanäppäimet näppäimistöllä.



Kuvio 13. Muotoiltu näkymä.

## 6.2 Tekniikoiden valinta

Työn toteutus aloitettiin ottamalla selvää eri tavoista siirtää selaingrafiikkaa Oculus Riftiin näkyväksi. Ensimmäinen kokeiltu tapa oli Webrift-niminen Websocket-palvelin sekä Javascript-kirjasto, joka sallisi Oculus Riftin käytön web-sovelluksissa. Ongelma tässä oli kuitenkin se, että olisi käyttäjien kannalta hankalaa, jos jokaisen tarvitsisi pystyttää oma Websocket-palvelin sekä ladata sovellus käyttöä varten. Oli etsittävä jokin suurempi ratkaisu.

Toisena vaihtoehtona oli luoda itse renderöitävän Canvaksen päälle raamit, jotka imitoisivat Oculus Riftin kuvakulman toimintaa. Oculus Rift asetettaisiin "Extended mode" -tapaan, jolloin se toimisi tavallaan toisena näyttönä. Tämän jälkeen web-sivu, jossa Canvas on ja joka halutaan saada näkymään kolmiulotteisena, tuotaisiin manuaalisesti toiselle näytölle eli Oculus Riftin näytölle ja aseteltaisiin tarkasti, jotta kolmiulotteisen kuvan rakentaminen olisi sulavaa. Tämä ei kuitenkaan ollut hyvä ratkaisu sillä se vaatii paljon säätämistä sekä olisi hyvin hankala aloittelijoille.

Seuraavana kokeiltiin Brandon Jones -nimisen GitHub-käyttäjän luomaa WebVR-enabled Nightly Chromium Build -tapaan (Jones 2014). Tämä salli HTML5 Canvas-elementin renderöinnin Oculus Riftin näyttöön suoraan ilman että käyttäjän tarvitsee ladata mitään ylimääräistä. Riitti että ajettavassa ohjelmassa oli integroituna VRRenderer.js-niminen Javascript-kirjasto, joka mahdollisti kuvan jaon ja muotoilun sekä kameroiden toiminnot Oculus Riftiin sopivalla tavalla. Tämä osoitautui parhaaksi tavaksi, sillä vr-yhteensopivuus on tulossa osaksi suurimpia selaimia, kun sitä saadaan vähän optimoitua paremmaksi sekä tietoturvaa parannettua.

### 6.3 Toteutus

Työn toteutus alkoi liittämällä VRRenderer.js-kirjasto projektiin mukaan (kuvio 14).

```
1 <html>
2   <head>
3     <script src="VRRenderer.js"></script>
4   </head>
5 </html>
```

Kuvio 14. Kirjaston lisäys projektiin.

Seuraavaksi luotiin funktio "vrDeviceCallback", jossa alussa tarkistetaan olemassa olevat vr-laitteet ja tunnistetaan ensimmäinen vastaan tuleva virtuaalipäähine. Seuraavaksi haetaan sensorit, jotka ovat yhteydessä löydettyyn virtuaalilaitteeseen. Lopuksi tarkistetaan kummalla tällä hetkellä tuetuista selaimista funktio suoritettiin ja palautetaan sitä vastaavat arvot (kuvio 15).

```

1  function vrDeviceCallback(vrdevs) {
2      for (var i = 0; i < vrdevs.length; ++i) {
3          if (vrdevs[i] instanceof HMDVRDevice) {
4              vrHMD = vrdevs[i];
5              break;
6          }
7      }
8      for (var i = 0; i < vrdevs.length; ++i) {
9          if (vrdevs[i] instanceof PositionSensorVRDevice &&
10             vrdevs[i].hardwareUnitId == vrHMD.hardwareUnitId) {
11              vrHMDSensor = vrdevs[i];
12              break;
13          }
14      }
15  }
16
17  if (navigator.getVRDevices) {
18      navigator.getVRDevices().then(vrDeviceCallback);
19  } else if (navigator.mozGetVRDevices) {
20      navigator.mozGetVRDevices(vrDeviceCallback);
21  }

```

Kuvio 15. Virtuaalilaitteen haku ja tunnistus.

Seuraavaksi luodaan three.js-kirjastoa apuna käyttäen WebGL-rendereri sekä vr-rendereri, joka käyttää parametreina WebGL-rendereriä sekä laitetta johon kuva renderöidään. Sitten määritellään Canvas, joka renderöidään. Tässä tapauksessa se oli ensimmäinen luotu Canvas. Lopuksi, riippuen selaimesta, siirretään Canvas virtuaalilaseihin koko ruudulle, kun käyttäjä kutsuu sitä (kuvio 16).

```

1
2   renderer = new THREE.WebGLRenderer();
3   vrrenderer = new THREE.VRRenderer(renderer, vrHMD);
4   var renderCanvas = document.getElementsByTagName("canvas")[0];
5
6   if (renderCanvas.mozRequestFullScreen) {
7     renderCanvas.mozRequestFullScreen({
8       vrDisplay: vrHMD
9     });
10  } else if (renderCanvas.webkitRequestFullscreen) {
11    renderCanvas.webkitRequestFullscreen({
12      vrDisplay: vrHMD,
13    });
14  }

```

Kuvio 16. Rendereiden luonti.

Lopuksi tulee vielä renderöintifunktio eli kuvan päivitys. Aluksi pyydetään animaattorunkoa render-funktiolle. Tämä mahdollistaa mekanismin käytettävälle renderöintisilmukalle. Sitten haetaan sensorien avulla suunta, mihin virtuaalilasit osoittavat. Lopuksi kaikki renderöidään sekä tietokoneen näytöllä että virtuaalilaseissa (kuvio 17).

```

1  function render() {
2    requestAnimationFrame(render);
3    var state = vrHMDSensor.getState();
4    camera.quaternion.set(state.orientation.x,
5                          state.orientation.y,
6                          state.orientation.z,
7                          state.orientation.w);
8    vrrenderer.render(scene, camera);
9  }

```

Kuvio 17. Renderöinti.

Kun kuva saatiin renderöityä laselle ja seuraamaan liikettä, pystyi ohjelmaa alkaa muotoilemaan Oculus Riffeille sopivaksi.

## 6.4 Ongelmat

Työn edetessä tuli vastaan paljon ongelmia sekä risteävyyksiä jo olemassa olevan koodin kanssa. Näiden korjaamiseen ja yhteensopivuuden luomiseen meni runsaasti aikaa. Ensimmäinen ongelma oli katselukulman liikuttamisen kanssa. Oculus Rift ottaa katselukulmien säädön käyttöön, joten nuolinäppäimillä ei voinut kääntää hahmoa. Ohessa on esimerkkiratkaisu, jossa luotiin ylimääräinen vektori, joka määrää kameran suunnan. Tämän vektorin suunta määräytyy sovelluksen hahmon kameran suunnan ja virtuaalilasien kvaternion ristitulona (kuvio 18). Työn toteuttamisessa käytetty ratkaisumalli oli hieman erilainen ja se ei valmistunut tämän työn kansitukseen mennessä.

```

2
3 function OrientationUpdated(Oculus_quaternion) {
4
5     var bodyAxis = new THREE.Vector3(0, 1, 0);
6
7     // Kvaternion hahmon kameran suunnasta Y-akselilla.
8     var bodyQuat = new THREE.Quaternion();
9     bodyQuat.setFromAxisAngle(bodyAxis, human_angle);
10
11     // Kvaternion virtuaalilasien nykyisestä suunnasta.
12     var riftQuat = new THREE.Quaternion(Oculus_quaternion.x, Oculus_quaternion.y,
13                                         Oculus_quaternion.z, Oculus_quaternion.w);
14
15     // Kerrotaan hahmon kameran ja virtuaalilasien kvaternionit keskenään.
16     bodyQuat.multiply(riftQuat);
17
18     // Uusi vektori osoittamaan Z-akselia, jota pyöritetään
19     // riippuen hahmon kameran ja virtuaalilasien kulmasta.
20     var xzVector = new THREE.Vector3(0, 0, 1);
21     xzVector.applyQuaternion(bodyQuat);
22
23     // Lasketaan X/Z-akselin kulma riippuen hahmon kameran ja virtuaalilasien kulmasta.
24     viewAngle = Math.atan2(xzVector.z, xzVector.x) + Math.PI;
25
26     // Kameran päivitys osoittamaan uuteen suuntaan.
27     camera.quaternion.copy(bodyQuat);
28 }
29

```

Kuvio 18. Kameran kontrollointi



Kun kameran kontrollointi on luotu, voi sovelluksessa liikkumisen toteuttaa esimerkiksi seuraavalla tavalla (kuvio 19).

```
2
3 function Movement(deltaTime) {
4
5     var step      = 25 * deltaTime; // 25 yksikköä sekunnissa
6     var turn_speed = (55 * deltaTime) * Math.PI / 180; // 55 astetta sekunnissa
7
8     // Eteen / Taakse
9     if(keys[FORWARD]){
10         human_position.x += Math.cos(viewAngle) * step;
11         human_position.z += Math.sin(viewAngle) * step;
12     }
13     if(keys[BACKWARD]){
14         human_position.x -= Math.cos(viewAngle) * step;
15         human_position.z -= Math.sin(viewAngle) * step;
16     }
17
18     // Kääntyminen
19     if(keys[TURN_RIGHT]){
20         human_angle += turn_speed;
21     }
22
23     if(keys[TURN_LEFT]){
24         human_angle -= turn_speed;
25     }
26 }
27
```

Kuvio 19. Liikkuminen

Muita huomioitavia seikkoja oli pintojen korostuksen poisto, kun siirrytään vr-tilaan. Tämä aiheutti visuaalista epämukavuutta lasella. Tämän korjaaminen onnistui ohittamalla tietty haara koodia, joka luo korostukset, jos siirrytään vr-tilaan. Seuraava ongelma oli jaettu näkymä. Tämä ei sopinut virtuaalilaseihin, joten kyseinen näkymä ohitetaan, jos vr-tila on käytössä. Lisäksi havaittiin, että anti-aliasing käyttöönnotto toi huomattavasti sulavuutta toimintaan. Tämä poisti turhan ”väreilyn” ympäristöstä.

Ongelma syntyi myös, kun palattiin vr-tilasta takaisin normaaliin näkymään. Kuva jäi puolitetuksi, joten tähän tuli myös keksiä keino palauttaa näkymä kokonaiseksi yhdeksi ruuduksi. Ratkaisu löytyi, kun tutki virtuaaliefektin luomisessa käytettävää apukirjastoa. Kun virtuaalitilaan siirryttiin, saksittiin kuva keskeltä poikki kahdeksi omaksi ruuduksi. Tämän efektin kumoamiseksi tuli luoda komento, joka suoritetaan, kun poistutaan kokoruututilasta. Komennolla määrätään saksittujen ruutujen määrä nolaksi sekä luodaan ainoasta Canvas-ikkunasta koko selainikkunan kokoinen. (Terrel 2014b.)



## 7 YHTEENVETO JA POHDINTA

Opinnäytetyön tuloksena saatiin luotua melko helppo ja nopea tapa muuntaa Google Chromen tai Mozilla Firefoxin HTML5 Canvas-elementillä toteutettu näkymä siirrettäväksi Oculus Rift -laseihin, sekä muunneltua se luomaan oikeanlaisen 3D-ympäristön Oculus Rift Development Kit 2 -virtuaalilaseilla. Tämä tapa on myös erittäin käyttäjäystävällinen, kun vr-tuki tulee osaksi käyttäjille suunnattuja selaimia. Virtuaalitilaan siirtyminen ja sieltä palaaminen saattaa aiheuttaa ongelmia riippuen sovelluksesta. Tässä työssä ongelmia syntyi melko paljon, koska sovellus on monipuolinen ja interaktiivinen. Kuitenkin suurin osa ongelmista saatiin ratkaistua tämän työn valmistumiseen mennessä.

Tämä työ luotiin esimerkksiovelluksella ja näkisin sen olevan toteutettavissa myös monessa muussa web-sovelluksessa, joista halutaan luoda virtuaalilaseilla toimiva kolmiulotteinen näkymä. Työn toteutukseen kuluvat kustannukset voivat olla olemattomat, mutta olisi kuitenkin suositeltavaa hankkia virtuaalilasit toimivuuden tarkistamisen vuoksi.

Työn luominen vaati osaamisen web-kielten ja selaimien toiminnasta sekä tietokonekokoonpanojen toteuttamisesta. Näytönohjaimen on oltava tarpeeksi tehokas, lisäksi ajurien on oltava päivitettyinä. Oculus Rift tarvitsee kaksi USB 2.0 -paikkaa sekä yhden HDMI 1.4 -paikan. Oculus Rift tarvitsee myös ajurit, jotka ovat ladattavissa sen kotisivulta, sekä sovelluksen, jolla voi säätää sisäisiä asetuksia, kuten esimerkiksi silmien etäisyys linseistä ja käyttäjän pituus.

Sovellus ei osaa automaattisesti tarkistaa linssien etäisyyttä näytöstä, vaan se tulee manuaalisesti säätää ohjelmistokehitystyökalulla. Kuva tarkentuu, jos asetuksen säätää oikein, sillä sovelluksessa on dynaaminen chroman korjaus. (Oculus User Guide 2012.)

Käyttäjän pituuden säätö puolestaan vaikuttaa sovelluksissa käyttäjän hahmon korkeuteen, sekä kameroiden korkeuteen. Tämän tarkoituksena on tuoda koon ja ruumiillistuman tunnetta. (Oculus User Guide 2012.)

Liikkuminen sovelluksessa käyttämällä näppäimistöä voi olla hankalaa niille, jotka eivät ole pelanneet paljoa tietokonepelejä. Virtuaalilasit päässä on hankala yrittää löytää tarvittavat näppäimet näppäimistöltä. Tälle oli tarkoituksena kehitellä käyttäjäystävällisempi tapa. Yksi vaihtoehto oli hiiren tuonti osaksi kameran kääntelyä, jolloin hahmolla kuljettaisiin siihen suuntaa johon kamera osoittaa. Toinen idea oli integroida xbox 360 -ohjain osaksi liikkumismenetelmiä. Kuitenkaan aika ei riittänyt kyseisten menetelmien testaamiseen.

## LÄHTEET

- Ajax. Ei päiväystä. What is AJAX?. [www-dokumentti]. tutorialspoint. [Viitattu 17.2.2015]. Saatavissa: [http://www.tutorialspoint.com/ajax/what\\_is\\_ajax.htm](http://www.tutorialspoint.com/ajax/what_is_ajax.htm)
- Apache friends. 2015. what is XAMPP?. [www-dokumentti]. Apache friends. [Viitattu 8.3.2015]. Saatavissa: <https://www.apachefriends.org/index.html>
- Boesch, F. 2013. Why you should use WebGL. [www-dokumentti]. codeflow. [Viitattu 17.2.2015]. Saatavissa: <http://codeflow.org/entries/2013/feb/02/why-you-should-use-webgl/>
- Brutzman, D. 1998. The Virtual Reality Modeling Language and Java. [www-dokumentti]. Naval Postgraduate School California. [Viitattu 16.2.2015]. Saatavissa: <http://www.web3d.org/x3d/content/examples/Basic/ScriptConformance/VrmlJavaPaper.pdf>
- Crockford, D. 2001. Javascript: The World's Most Misunderstood Programming Language. [www-dokumentti]. Crockford. [Viitattu 17.2.2015]. Saatavissa: <http://javascript.crockford.com/javascript.html>
- Digitalized technologies. Ei päiväystä. The omni is the missing link, and also healthy! - the future of HCI. [www-dokumentti]. Digitalized technologies. [Viitattu 5.3.2015]. Saatavissa: <https://digitalizedtechnologies.wordpress.com/the-omni-will-improve-your-life-the-future-of-hci/>
- Evolution of the web. 2012. The Evolution of the Web. [www-sivu]. evolutionoftheweb. [Viitattu 16.2.2015]. Saatavissa: <http://www.evolutionoftheweb.com/?hl=en#/evolution/day>
- Grubb, J. 2014. Everything you need to know about the oculus rift. [www-dokumentti]. VentureBeat. [Viitattu 16.2.2015]. Saatavissa: <http://venturebeat.com/2014/08/06/everything-you-need-to-know-about-the-oculus-rift/>
- Hakala, H. 2014. Virtuaalitodellisuus teollisuudessa - Lisätyn todellisuuden mahdollisuudet. [www-dokumentti]. Cybercom Group. [Viitattu 26.1.2015]. Saatavissa: <http://www.cybercom.com/fi/Suomi/Yritys/Blogit/Blogit/virtuaalitodellisuus-teollisuudessa/>
- Hietala, L. 2014. Virtuaalitodellisuus. [www-dokumentti]. SPACEBIMBOM. [Viitattu 26.1.2015]. Saatavissa: <http://spacebimbom.com/kategoriat/internet/virtuaalitodellisuus.php>

- iFixit. 2014. Oculus Rift Development Kit 2 Teardown. [www-dokumentti]. iFixit. [Viitattu 5.3.2015]. Saatavissa: <https://www.ifixit.com/Teardown/Oculus+Rift+Development+Kit+2+Teardown/27613>
- Jones, B. 2014. Bringing VR to Chrome. [www-dokumentti]. TojiCode. [Viitattu 17.2.2015]. Saatavissa: <http://blog.tojicode.com/2014/07/bringing-vr-to-chrome.html>
- jQuery. Ei päiväystä. jQuery - write less, do more. [www-sivu]. jQuery Foundation. [Viitattu 17.2.2015]. Saatavissa: <http://jquery.com/>
- Kamm, D. 2012. Army's virtual reality plan: A digital doppelganger for every soldier. [www-dokumentti]. Wired. [Viitattu 5.3.2015]. Saatavissa: <http://www.wired.com/2012/01/army-virtual-reality/>
- Kickstarter. 2012. Oculus Rift. [www-sivu]. Oculus. [Viitattu 16.2.2015]. Saatavissa: <https://www.kickstarter.com/projects/1523379957/oculus-rift-step-into-the-game>
- Lepton, K. 2011. Future Virtual Reality. [www-dokumentti]. Future Technology 500. [Viitattu 26.1.2015]. Saatavissa: <http://www.futuretechnology500.com/index.php/future-virtual-reality/>.
- Makzan. 2011. HTML5 Games Development by Example - Beginner's Guide. Birmingham, UK. Packet Publishing Ltd.
- Melnikov, A. 2011. The WebSocket Protocol. [www-dokumentti]. Internet FAQ Archieves. [Viitattu 17.2.2015]. Saatavissa: <http://www.faqs.org/rfcs/rfc6455.html>
- Mozilla Developer Network. 2014. CSS3. [www-dokumentti]. Mozilla. [Viitattu 17.2.2015]. Saatavissa: <https://developer.mozilla.org/en-US/docs/Web/CSS/CSS3>
- Mr Buzz. 2015. shoot em up project - an opengl game for Linux and Windows. [www-sivu]. Youtube. [Viitattu 5.3.2015]. Saatavissa: <https://www.youtube.com/watch?v=2rc3h4mYBx8>
- mrdoob. 2013. Getting Started. [www-dokumentti]. GitHub. [Viitattu 17.2.2015]. Saatavissa: <https://github.com/mrdoob/three.js/wiki/Getting-Started>
- mrdoob. 2015. Contributors. [www-sivu]. Github. [Viitattu 17.2.2015]. Saatavissa: <https://github.com/mrdoob/three.js/graphs/contributors>
- Mähönen, M. 2015. Henkilökohtainen tiedonanto. Chief Operation Officer. 3D Avenue International Oy.

- Oculus VR. 2012. Oculus User Guide. [www-dokumentti]. Oculus VR. [Viitattu 9.3.2015]. Saatavissa: [http://static.oculus.com/sdk-downloads/documents/Oculus\\_User\\_Guide\\_0.4.4.pdf](http://static.oculus.com/sdk-downloads/documents/Oculus_User_Guide_0.4.4.pdf)
- OpenGL. 2015. History of OpenGL. [www-dokumentti]. The Industry's Foundation for High Performance Graphics. [Viitattu 17.2.2015]. Saatavissa: [https://www.opengl.org/wiki/History\\_of\\_OpenGL](https://www.opengl.org/wiki/History_of_OpenGL)
- php. Ei päiväystä. What is PHP?. [www-dokumentti]. The PHP Group. [Viitattu 17.2.2015]. Saatavissa: <http://php.net/manual/en/intro-what-is.php>
- Prall, C. Ei päiväystä. Physijs. [www-sivu]. GitHub. [Viitattu 17.2.2015]. Saatavissa: <http://chandlerprall.github.io/Physijs/>
- ScienceDaily. Ei päiväystä. 3D computer graphics. [www-dokumentti]. ScienceDaily. [Viitattu 27.1.2015]. Saatavissa: [http://www.sciencedaily.com/articles/0/3d\\_computer\\_graphics.htm](http://www.sciencedaily.com/articles/0/3d_computer_graphics.htm)
- Selim. 2011. WEBSOCKETS VS AJAX. [www-dokumentti]. geeksdraft. [Viitattu 17.2.2015]. Saatavissa: <http://www.geeksdrafts.net/blog/2011/01/10/websockets-vs-ajax/>
- Slick, J. Ei päiväystä. Introducing the computers graphics pipeline. [www-dokumentti]. about tech. [Viitattu 16.2.2015]. Saatavissa: <http://3d.about.com/od/3d-101-The-Basics/tp/Introducing-The-Computer-Graphics-Pipeline.htm>
- Sitepoint. 2014. Introduction to the HTML5 Canvas Element. [www-dokumentti]. Sitepoint Team. [Viitattu 3.3.2015]. Saatavissa: <http://www.sitepoint.com/about-us/>
- Starr, M. 2014. Oculus VR buys hand-tracking expert Nimble VR. [www-dokumentti]. cnet. [Viitattu 16.2.2015]. Saatavissa: <http://www.cnet.com/news/oculus-vr-buys-hand-tracking-expert-nimble-vr/>
- Sucan, M. 2009. HTML 5 Canvas - the Basics. [www-dokumentti]. Opera Development Team. [Viitattu 17.2.2015]. Saatavissa: <https://dev.opera.com/articles/html5-canvas-basics/>
- Tarun. 2013. Mozilla and Epic Tap Power of WebGL. [www-dokumentti]. Nextpowerup. [Viitattu 5.3.2015]. Saatavissa: <http://www.nextpowerup.com/news/1056/mozilla-and-epic-tap-power-of-webgl.html>
- Telepresenceoptions. 2012. Virtual reality technology comes of age. [www-dokumentti]. Telepresenceoptions. [Viitattu 5.3.2015]. Saatavissa: [http://www.telepresenceoptions.com/2012/03/virtual\\_reality\\_technology\\_com/](http://www.telepresenceoptions.com/2012/03/virtual_reality_technology_com/)

- Terrel, R. 2014a. three-vr-renderer. [www-sivu]. GitHub. [Viitattu 17.2.2015]. Saatavissa: <https://github.com/wwwtyro/three-vr-renderer>
- Terrel, R. 2014b. VRRenderer.js. [www-dokumentti]. GitHub. [Viitattu 17.2.2015]. Saatavissa: <https://raw.githubusercontent.com/wwwtyro/three-vr-renderer/master/VRRenderer.js>
- THREEx. 2015. Extensions. [www-sivu]. THREEx Working group. [Viitattu 17.2.2015]. Saatavissa: <http://www.threejsgames.com/extensions/>
- Tironeac, G. 2013. The Pipeline Behind Modeling and Animating a Game Character in Zelgor. [www-dokumentti]. Assist-software. [Viitattu 5.3.2015]. Saatavissa: <http://assist-software.net/blog/pipeline-behind-modeling-and-animating-game-character-zelgor>
- Virtuix Omni. 2014. Frequently asked questions. [www-sivu]. Virtuix. [Viitattu 16.2.2015]. Saatavissa: <http://www.virtuix.com/frequently-asked-questions/>
- W3C. 2012. FAQs. [www-dokumentti]. [Viitattu 17.2.2015]. Saatavissa: <http://www.w3.org/html/wiki/FAQs>
- w3schools. Ei päiväystä. Javascript Tutorial. [www-dokumentti]. World Wide Web Consortium. [Viitattu 17.2.2015]. Saatavissa: <http://www.w3schools.com/js/>
- WebGL. 2011. Getting Started. [www-dokumentti]. WebGL public wiki. [Viitattu 17.2.2015]. Saatavissa: [https://www.khronos.org/webgl/wiki/Getting\\_Started](https://www.khronos.org/webgl/wiki/Getting_Started)
- WebSocket. 2013. About HTML5 WebSocket. [www-dokumentti]. Kaazing Corporation. [Viitattu 17.2.2015]. Saatavissa: <https://www.websocket.org/aboutwebsocket.html>
- WHATWG. Ei päiväystä. HTML: The Living Standard. [www-dokumentti]. Web Hypertext Application Technology Working Group. [Viitattu 17.2.2015]. Saatavissa: <https://developers.whatwg.org/the-canvas-element.html#the-canvas-element>
- Winn, R. 2013. What Does the Future Hold for JavaScript?. [www-dokumentti]. IZENDA. [Viitattu 17.2.2015]. Saatavissa: <http://www.izenda.com/blog/what-does-the-future-hold-for-javascript/>
- X3D. 2001. Frequently Asked Questions. [www-dokumentti]. X3D - 3D ANYWHERE. [Viitattu 16.2.2015]. Saatavissa: [http://www.siggraph.org/education/materials/HyperGraph/webgraphics/x3d/x3d\\_faq\\_files/](http://www.siggraph.org/education/materials/HyperGraph/webgraphics/x3d/x3d_faq_files/)

**LIITTEET**